

**Department of Computer Science and Engineering
KPR Institute of Engineering and Technology**



B.E. – COMPUTER SCIENCE AND ENGINEERING

LABORATORY RECORD

U19CS603 – CLOUD COMPUTING LABORATORY

(Regulation 2019)

MAY 2022



KPR INSTITUTE OF ENGINEERING AND TECHNOLOGY
(Autonomous)
COIMBATORE – 641 407

LABORATORY RECORD

Name :

Roll Number :

Subject Code & Title:

Department :

Year & Semester :

This is the certified record of work done by.....

Register Number.....

Staff In- Charge

Head of the Department

Place:

Date:

He/ She has submitted the record for the End Semester Practical
Examination held on

Internal Examiner

External Examiner

INDEX

Ex. No.	Date	Experiment Name	Page No	Marks	Sign
1		Install Virtualbox/VMware Workstation with different flavours of linux or windows OS on top of windows7 or 8.			
2		Install a C compiler in the virtual machine created using virtual box and execute Simple Programs			
3		Install Google App Engine. Create hello world app and other simple web applications using python/java.			
4		Use GAE launcher to launch the web applications.			
5		Create, Deploy and Launch Virtual Machines in OpenStack			
6		Creating a Warehouse Application in SalesForce.com.			
7		Install Hadoop single node cluster and run simple applications like Word Count			
8		Sign up for a free account in AWS , create and launch your VM in the EC2 Cloud and perform Web hosting in the VM			
9		Create a free account in IBM cloud and develop and IoT application with Node-RED			
10		Create a free account in Microsoft Azure cloud and develop a bot for business applications			
Total Marks					

Vision of the Institution

To become a premier institute of academic excellence by imparting technical, intellectual and professional skills to students for meeting the diverse needs of the industry, society, the nation and the world at large.

Mission of the Institution

1. Commitment to offer value-based education and enhancement of practical skills
2. Continuous assessment of teaching and learning process through scholarly activities
3. Enriching research and innovative activities in collaboration with industry and institute of repute
4. Ensuring the academic process to uphold culture, ethics and social responsibility

Vision of the Department

To foster the students by providing learner centric teaching environment, continuous learning, research and development to become thriving professionals and entrepreneurs to excel in the field of computer science and contribute to the society.

Mission of the Department

- Providing value-based education and contented learning experience to the students
- Educating the students with the state of art technologies and cultivating their proficiency in analytical and designing skills
- Enabling the students to achieve a successful career in Computer Science and Engineering or related fields to meet the changing needs of various stakeholders
- Guiding the students in research by nurturing their interest in continuous learning towards serving the society and the country

EXP NO: 1

DATE:

INSTALL VIRTUAL BOX / VMWARE WORKSTATION

AIM:

To Install Virtualbox/VMware Workstation with different flavours of linux or windows OS on top of windows7 or 8.

PROCEDURE:

Step 1- Download Link

Link for downloading the software is <https://www.vmware.com/products/workstation-pro/workstation-pro-evaluation.html>. Download the software for windows. Good thing is that there is no sign up process. Click and download begins. Software is around 541 MB.

Step 2- Download the installer file

It should probably be in the download folder by default, if you have not changed the settings in your browser. File name should be something like VMware-workstation-full-15.5.1-15018445.exe. This file name can change depending on the version of the software currently available for download. But for now, till the next version is available, they will all be VMware Workstation 15 Pro.

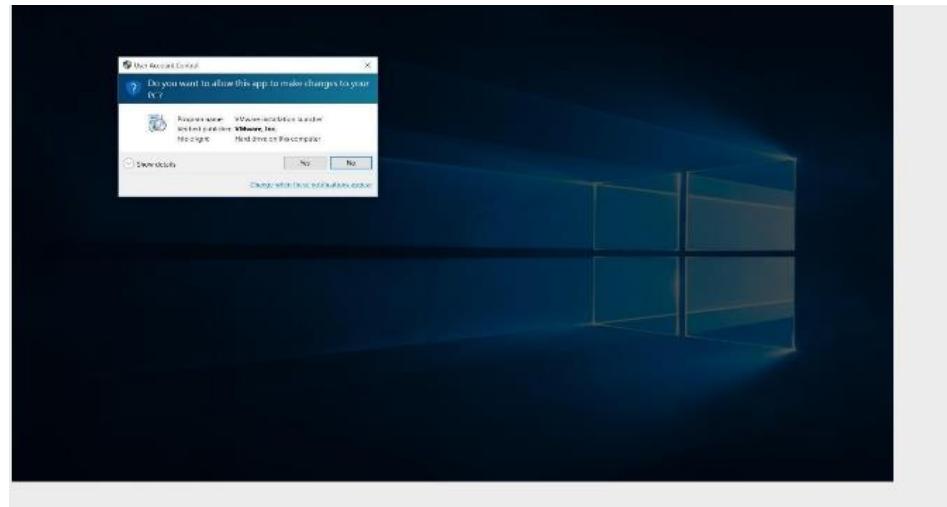
Step 3- Locate the downloaded installer file

For demonstration purpose, I have placed the downloaded installer on my desktop. Find the installer on your system and double click to launch the application.



Step 4- User Access Control (UAC) Warning

Now you should see User Access Control (UAC) dialog box. Click yes to continue. Initial Splash screen will appear. Wait for the process to complete.



Step 5- VMware Workstation Setup wizard

Now you will see VMware Workstation setup wizard dialog box. Click next to continue



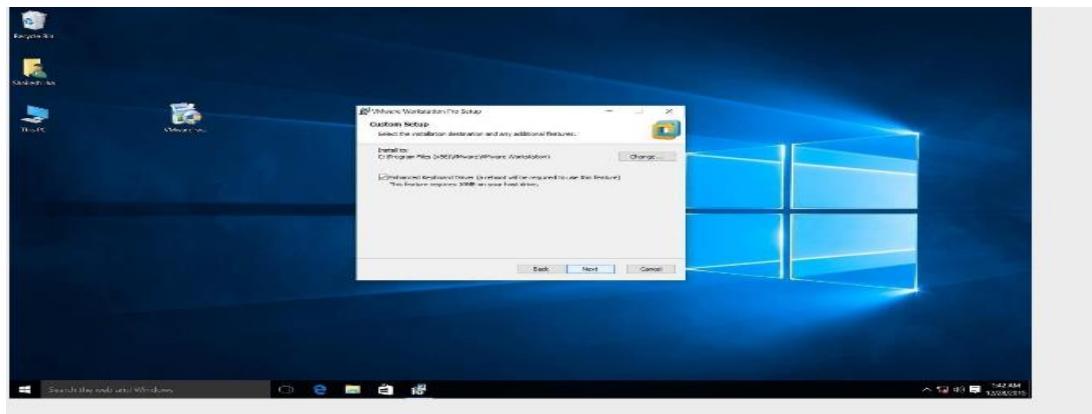
Step 6- End User Licence Agreement

This time you should see End User Licence Agreement dialog box. Check “I accept the terms in the Licence Agreement” box and press next to continue



Step 7- Custom Setup options

Select the folder in which you would like to install the application. There is no harm in leaving the defaults as it is. Also select Enhanced Keyboard Driver check box.

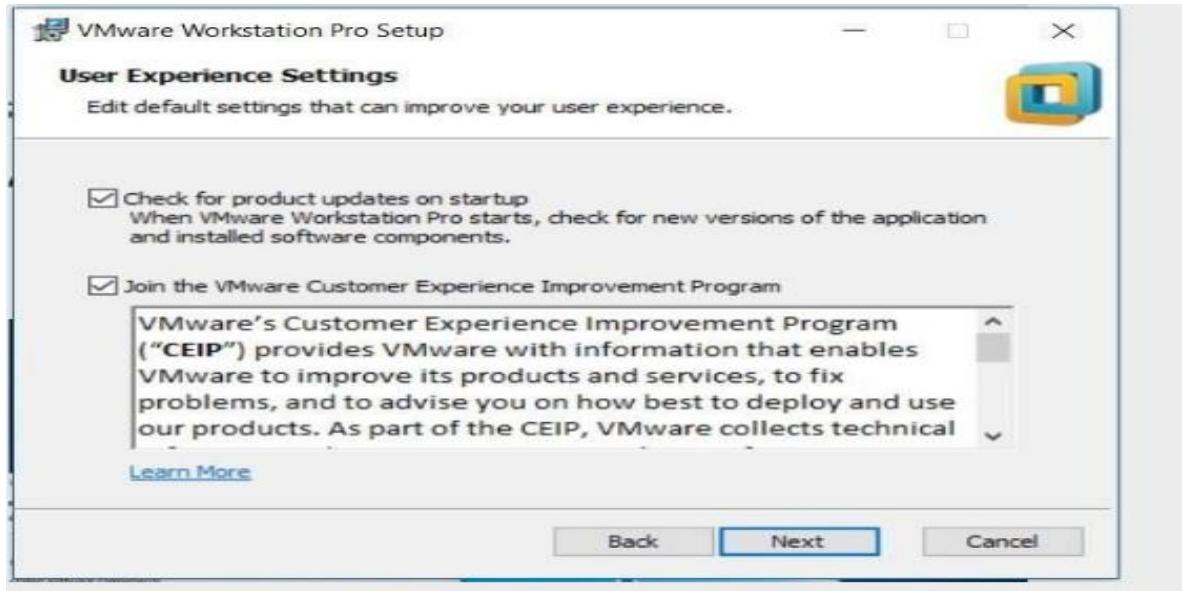


Step 8- User Experience Settings

Next you are asked to select “Check for Updates” and “Help improve VMware Workstation Pro”. Do as you wish. I normally leave it to defaults that is unchecked.

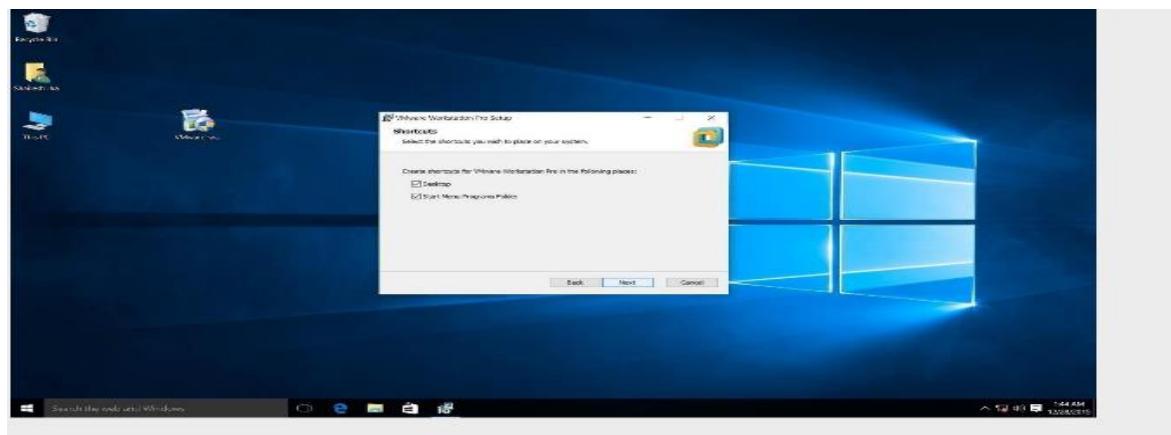
Step 9- Application Shortcuts preference

Next step is to select the place you want the shortcut icons to be placed on your system to launch the application. Please select both the options, desktop and start menu and click next.

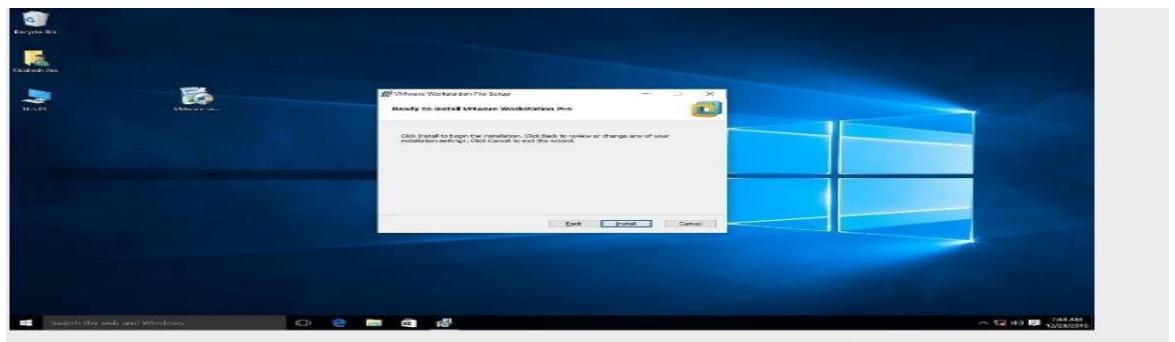


Step 10- Installation begins

Now you see the begin installation dialog box. Click install to start the installation process. Below screenshot shows Installation in progress. Wait for this to complete.

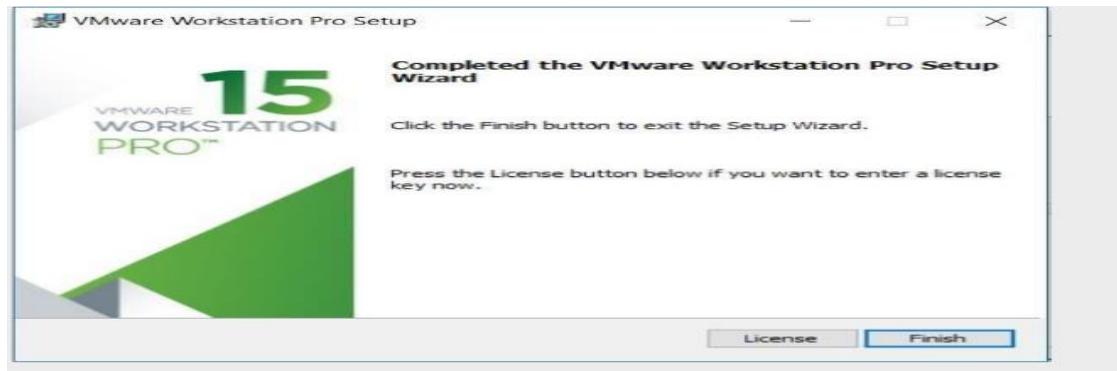


At the end you will see installation complete dialog box. Click finish and you are done with the installation process. You may be asked to restart your computer. Click on Yes to restart



Step 11- Launch VMware Workstation

After the installation completes, you should see VMware Workstation icon on the desktop. Double click on it to launch the application.



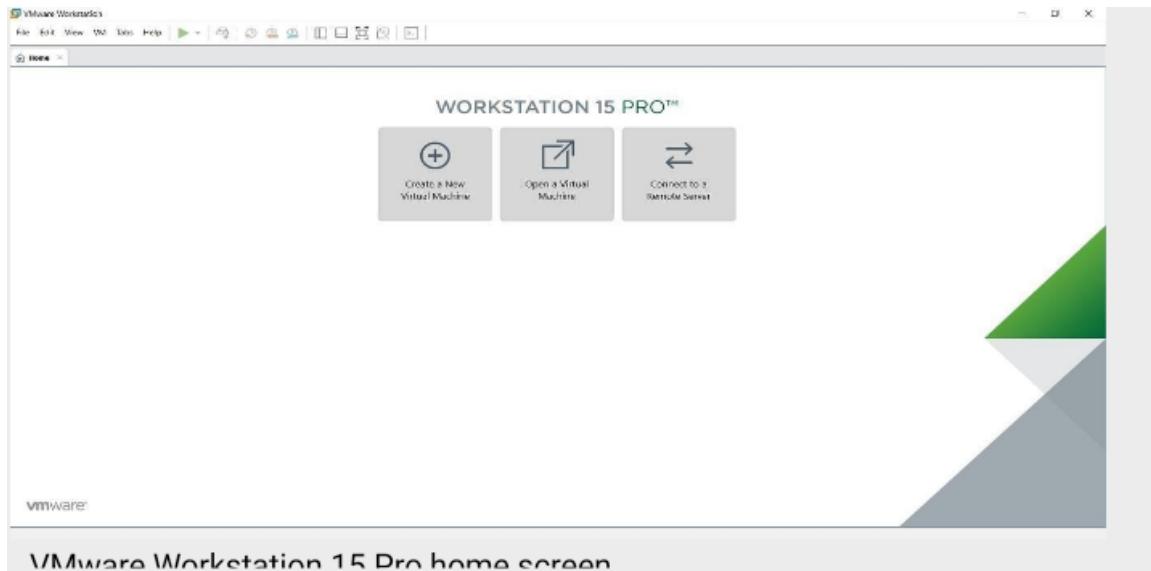
Step 12- Licence Key

If you see the dialog box asking for licence key, click on trial or enter the licence key. Then what you have is the VMware Workstation 15 Pro running on your windows 10 desktop. If don't have the licence key, you will have 30 days trial.

Step 13- At some point if you decide to buy

At some point of time if you decide to buy the Licence key, you can enter the Licence key by going to **Help- >Enter a Licence Key**. You can enter the 25 character licence key in the dialog box shown below and click OK. Now you have the licence version of the software.





VMware Workstation 15 Pro home screen

DEPARTMENT OF CSE		
Preparation(Algorithm)	4	
Observation(Program)	4	
Results(Output)	4	
Interpretation(Validation)	4	
Viva-Voce	4	
Total	20	

RESULT:

EXP NO: 2

DATE:

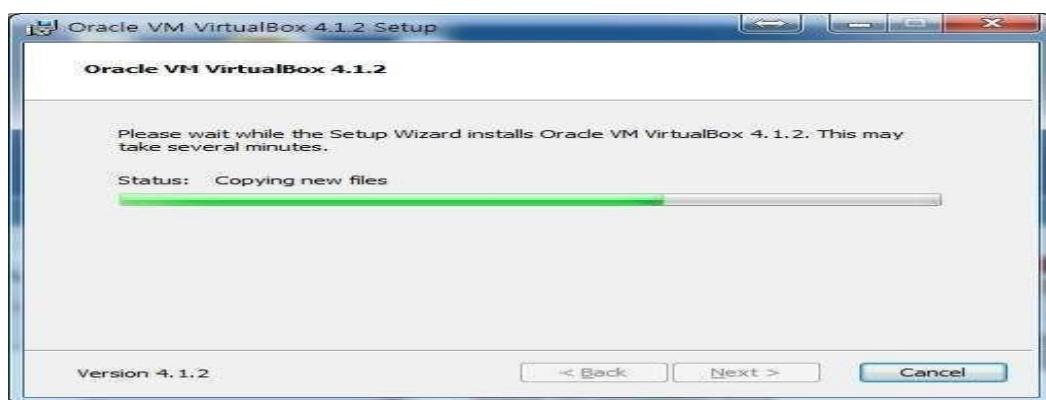
SETTINGUP C PROGRAMMING ENVIRONMENT

AIM:

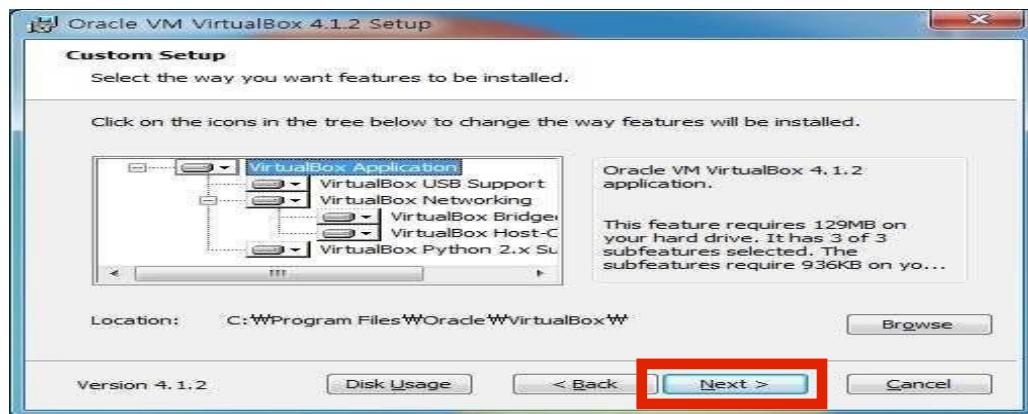
To write a C program in virtual machine

PROCEDURE:

- Visit <http://www.virtualbox.org/wiki/downloads>
- Download VirtualBox platform packages for your OS
- Open the Installation Package by double clicking
- Click continue and finish installing VirtualBox



- When finished installation, close the window.



- Visit the page <http://www.ubuntu.com/download/ubuntu/download> Choose the Latest version of Ubuntu and 32-bit and click Start Download"

1. Run VirtualBox by double-clicking the icon
2. Click "New" button on the top left corner

1

Download Ubuntu

Click the big orange button to download the latest version of Ubuntu. You will need to create a CD or USB stick to install Ubuntu.

Our long-term support (LTS) releases are supported for three years on the desktop. Perfect for organisations that need more stability for larger deployments.

Additional options

Take a look at a full list of our previous versions and alternative downloads >

Download options

Ubuntu 11.04 – Latest version

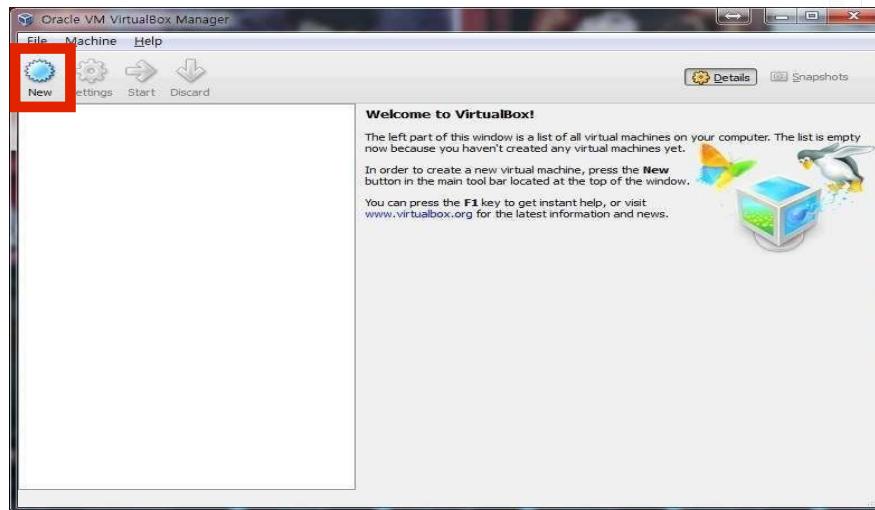
32-bit (recommended)

Download started

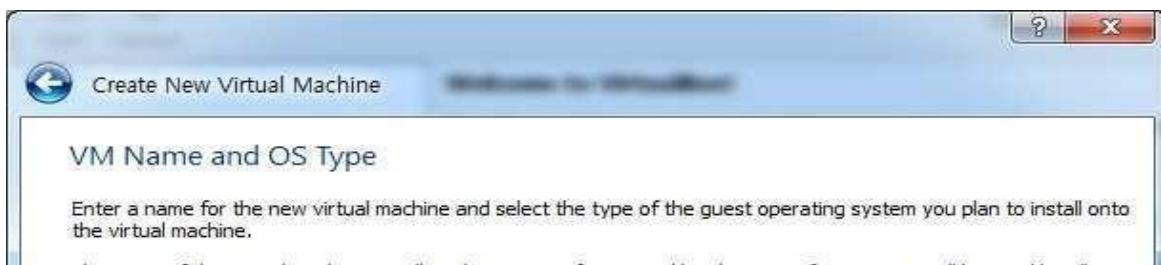
Ubuntu 11.04

32-bit

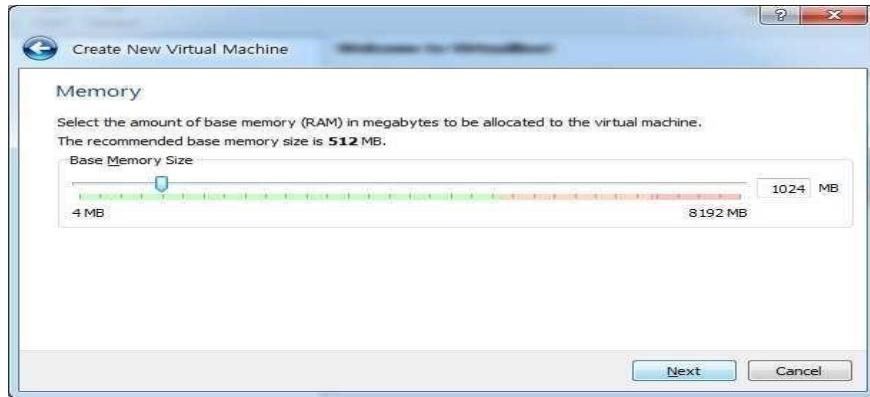
CLICK



3. Click “Continue” on the pop-up window
4. Type VMname, select “Linux” for the OS and choose “Ubuntu” for the version



5. Choose the amount of memory to allocate (suggest choosing between 512 MB to 1024MB)
6. Click Continue or Next



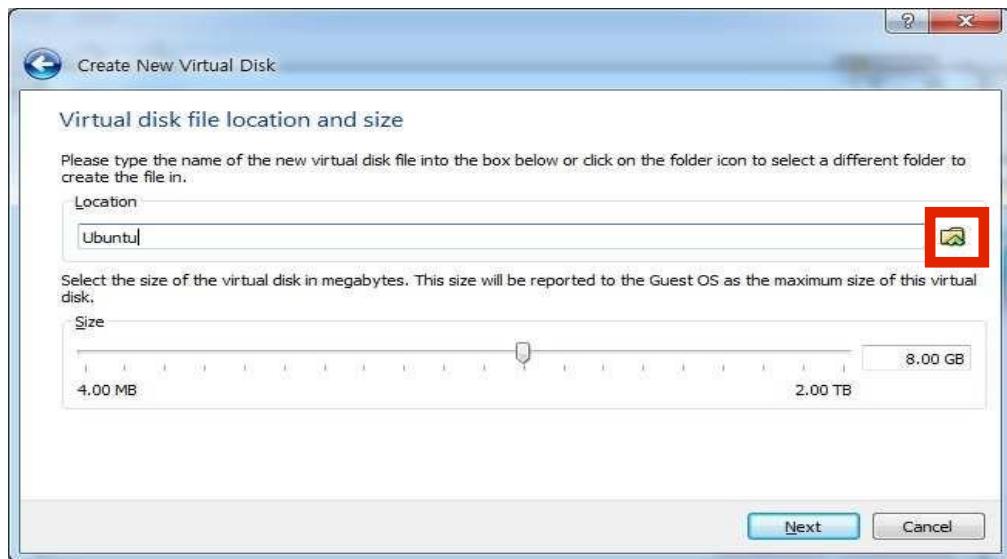
7. Choose create a new virtual harddisk
8. Click Continue or Next
9. Choose VDI (VirtualBox DiskImage)
10. Click Continue or Next



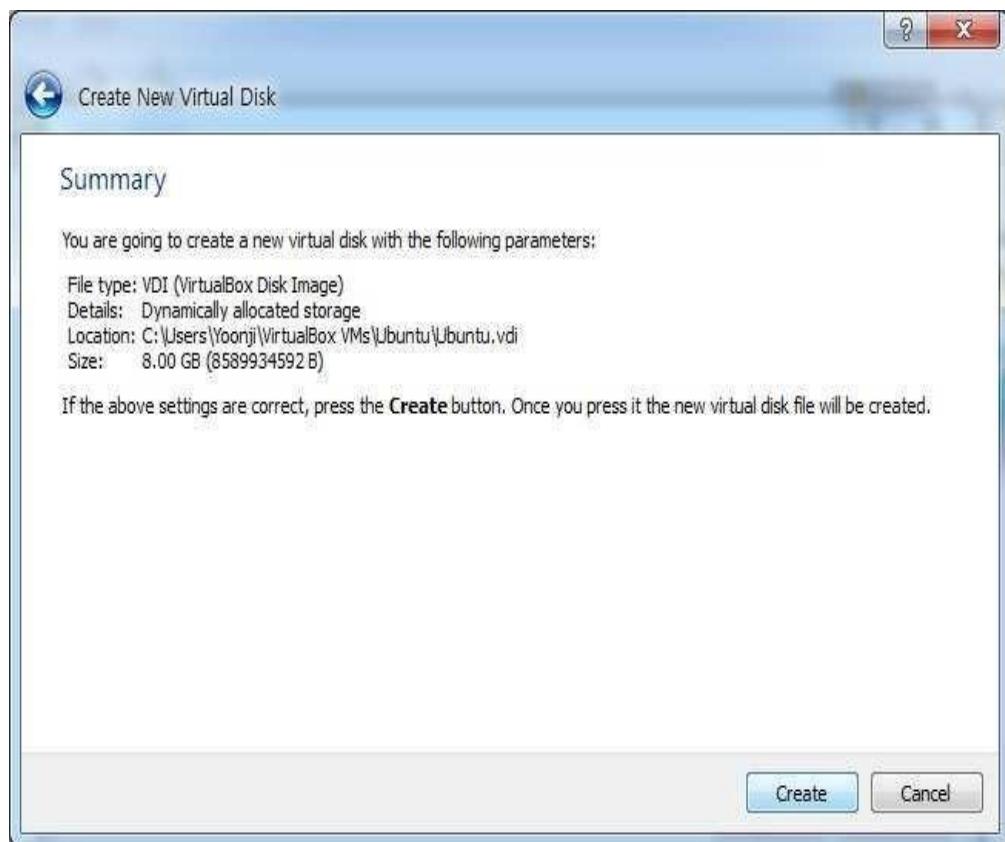
11. Choose “Dynamically Allocated” click continue. This way, the size of your Virtual Hard Disk will grow as you use.



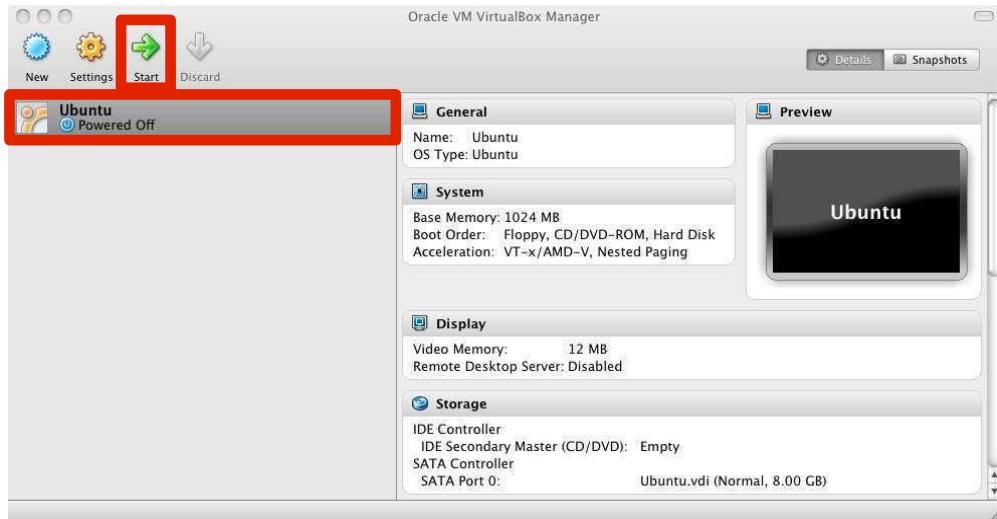
12. Click the folder icon and choose the ubuntu iso file you downloaded.
13. Select the size of the Virtual Disk (Ire command choosing 8 GB) and click continue



14. Click Create



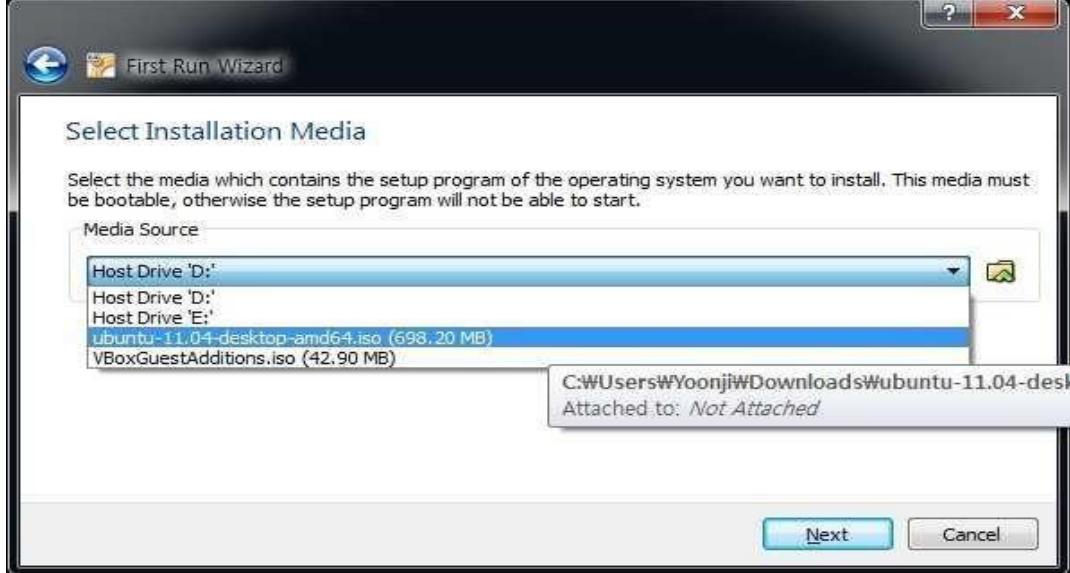
15. Choose Ubuntu from left column and click Start



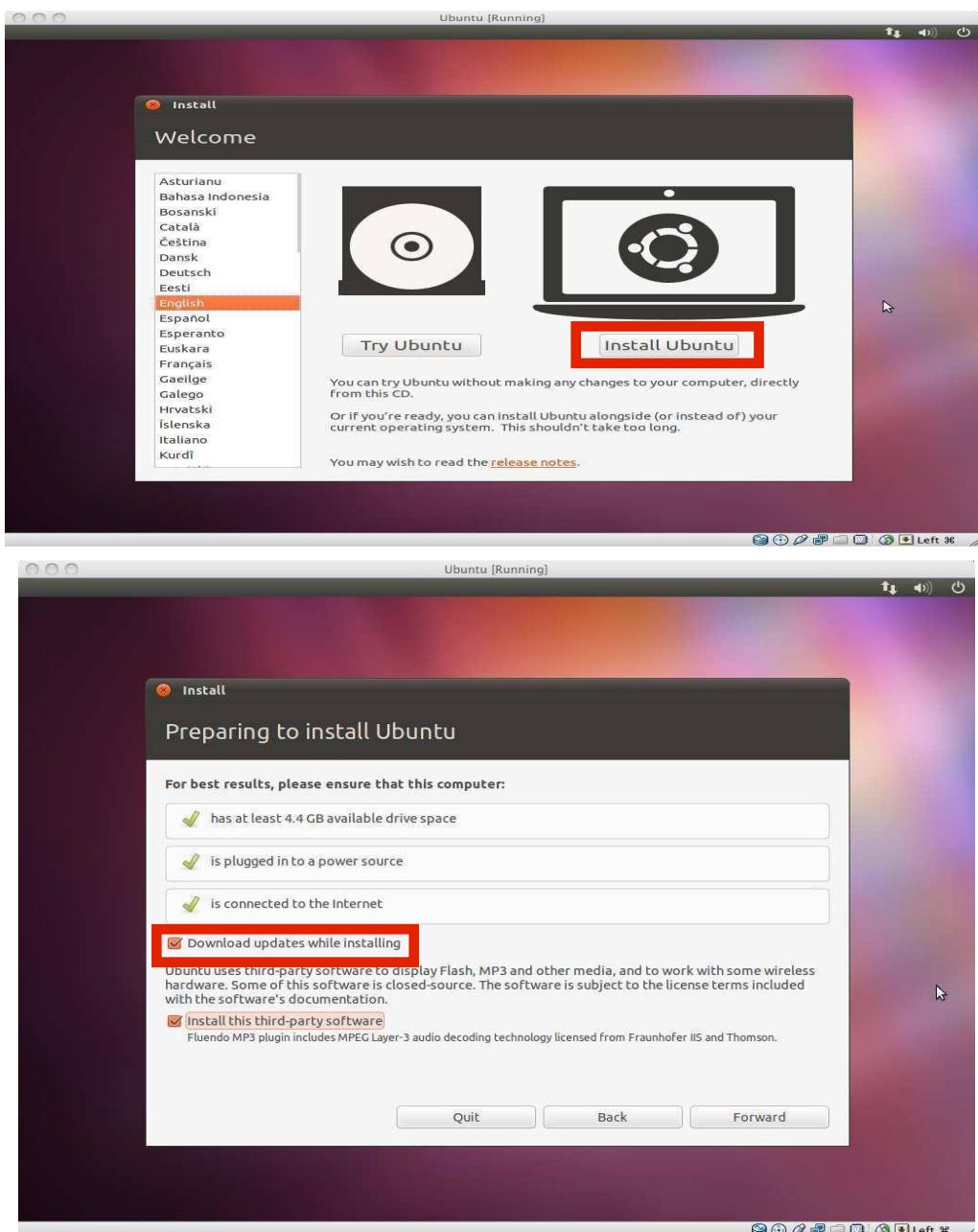
16. Click continue on pop-up window



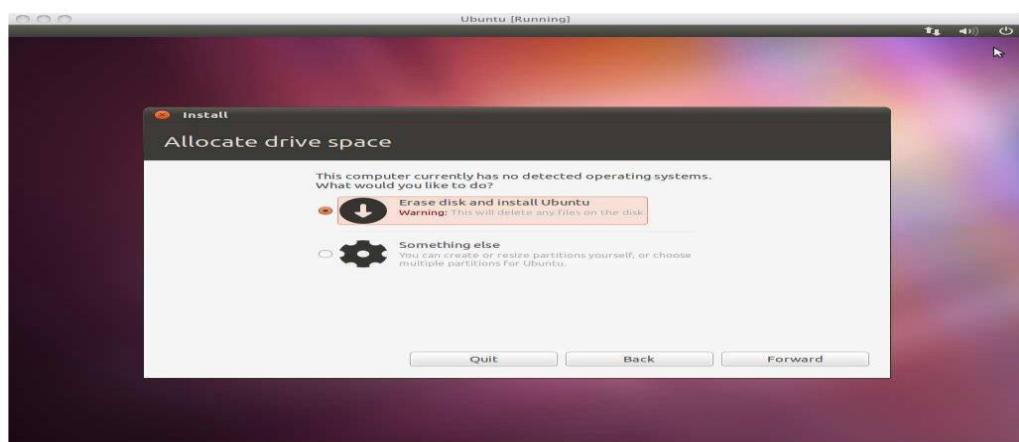
17. Click the folder icon and choose the ubuntu iso file youdownloaded and click continue and start
18. Click Install Ubuntu



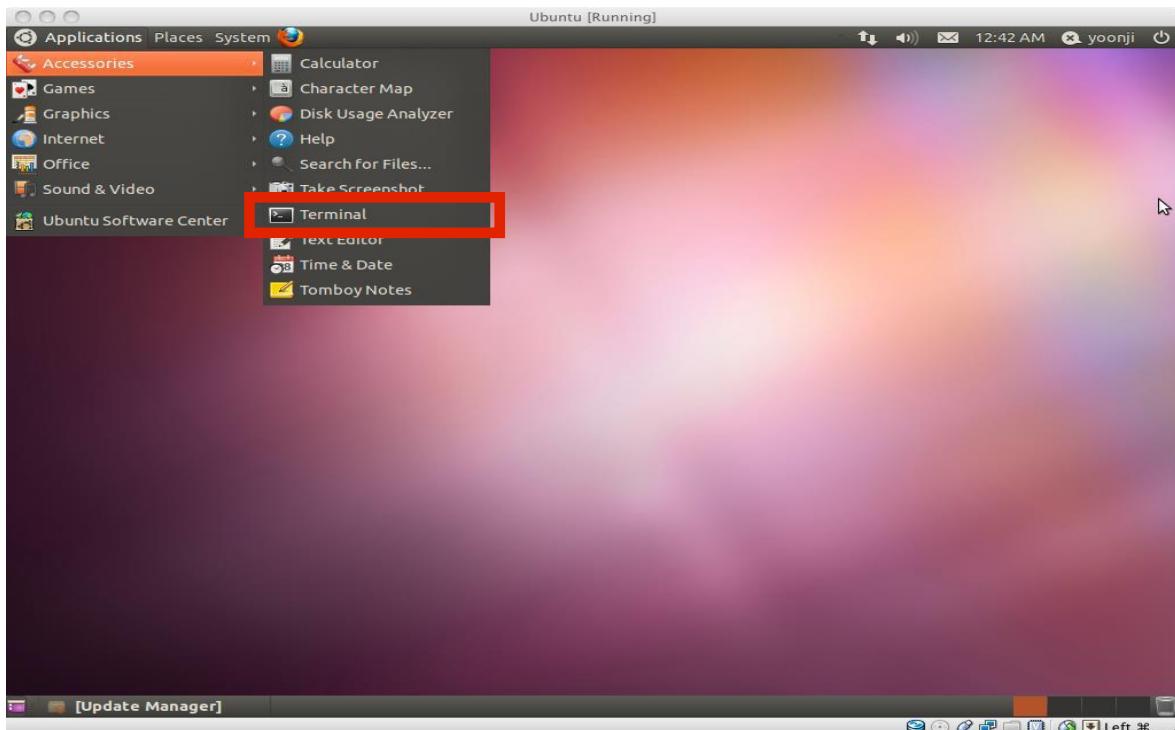
19. Check “Download updates” and click Forward
20. Choose “Erase disk and install ubuntu” and click forward



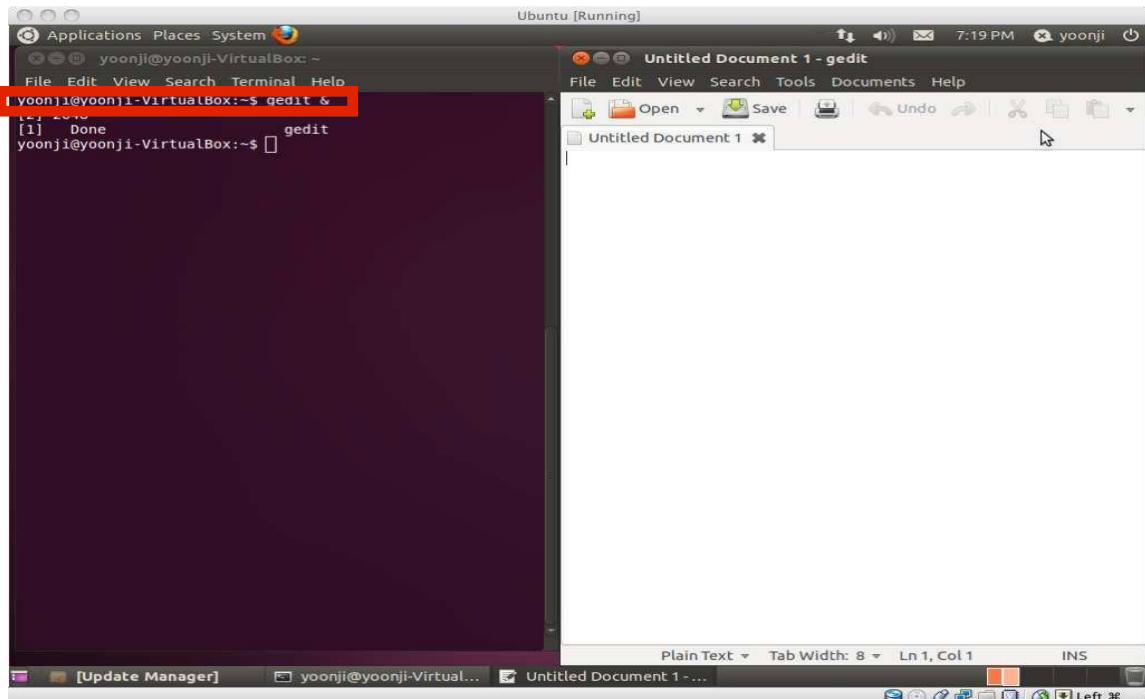
21. Click “Install Now” and wait. May be grab a snack.
22. When finished, click Restart and press Enter.



23. Open Terminal (Applications-Accessories-Terminal)

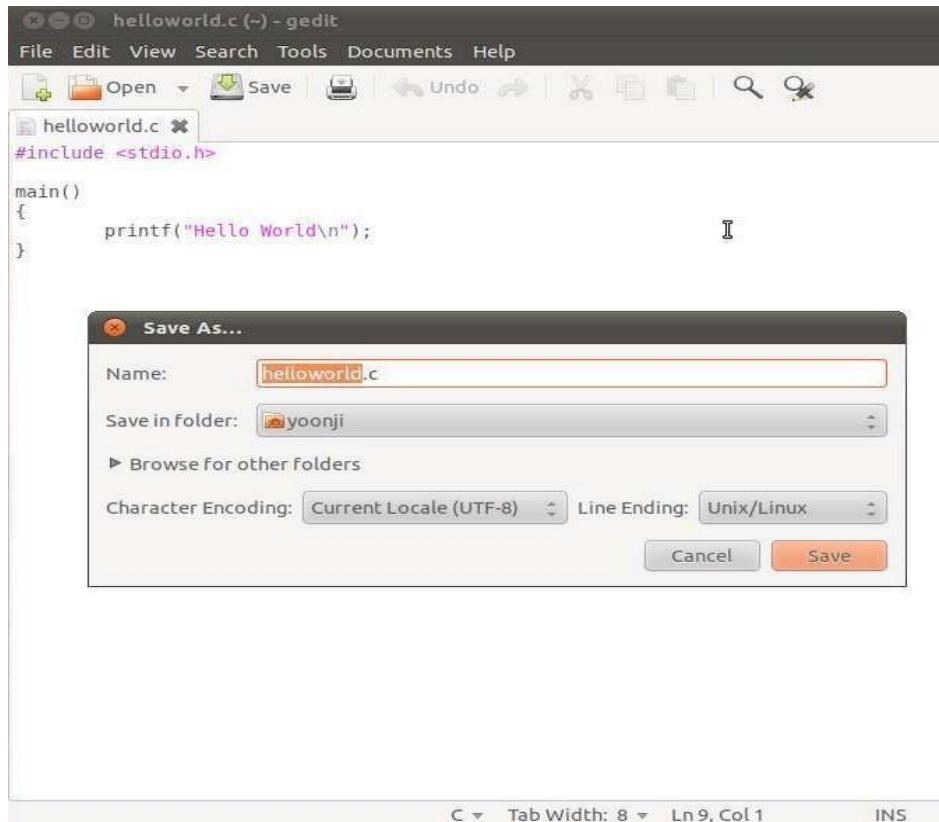


24. Open gedit by typing “gedit &” on terminal (You can also use any other Text Editor application)



25. Type the following on
gedit (or any other text
editor)
#include<stdio.h>

```
main()
{
    printf("Hello World\n");
}
```



26. Type “ls” on Terminal to see all file under current folder
27. Confirm that “helloworld.c” is in the current directory. If not, type cd DIRECTORY_PATH to go to the directory that has“helloworld.c”
28. Type “gcc helloworld.c” to compile, and type “ls” to confirm that an executable file “a.out” is created

```
yoonji@yoonji-VirtualBox:~$ ls
Desktop Downloads helloworld.c Pictures Templates
Documents examples.desktop Music Public Videos
yoonji@yoonji-VirtualBox:~$ gcc helloworld.c
yoonji@yoonji-VirtualBox:~$ ls
a.out Documents examples.desktop Music Public Videos
Desktop Downloads helloworld.c Pictures Templates
yoonji@yoonji-VirtualBox:~$
```

29. Type“./a.out”on Terminal to run the program
30. If you see “HelloWorld” on the next line, you just successfully ran your first C program!

```
yoonji@yoonji-VirtualBox:~$ ls
Desktop Downloads helloworld.c Pictures Templates
Documents examples.desktop Music Public Videos
yoonji@yoonji-VirtualBox:~$ gcc helloworld.c
yoonji@yoonji-VirtualBox:~$ ls
a.out Documents examples.desktop Music Public Videos
Desktop Downloads helloworld.c Pictures Templates
yoonji@yoonji-VirtualBox:~$ ./a.out
Hello World
yoonji@yoonji-VirtualBox:~$
```

DEPARTMENT OF CSE		
Preparation(Algorithm)	4	
Observation(Program)	4	
Results(Output)	4	
Interpretation(Validation)	4	
Viva-Voce	4	
Total	20	

RESULT:

EXP NO: 3

DATE:

INSTALL GOOGLE APP ENGINE

AIM:

Install Google App Engine. Create hello world app and other simple web applications using python/java.

PROCEDURE:

- This document describes the installation of the Google App Engine Software Development Kit (SDK) on a Microsoft Windows and running a simple “hello world” application.
- The App Engine SDK allows you to run Google App Engine Applications on your local computer.
- It simulates the run time environment of the Google App Engine infrastructure.

PRE REQUISITES: Python 2.5.4

If you don't already have Python 2.5.4 installed in your computer, download and Install Python 2.5.4 from:

<http://www.python.org/download/releases/2.5.4/>

DOWNLOAD AND INSTALL

Download the Google App Engine SDK

Before downloading, please read the [Terms](#) that govern your use of the App Engine SDK.

Please note: The App Engine SDK is under **active development**, please keep this in mind as you explore its capabilities. See the [SDK Release Notes](#) for the information on the most recent changes to the App Engine SDK. If you discover any issues, please feel free to notify us via our [Issue Tracker](#).

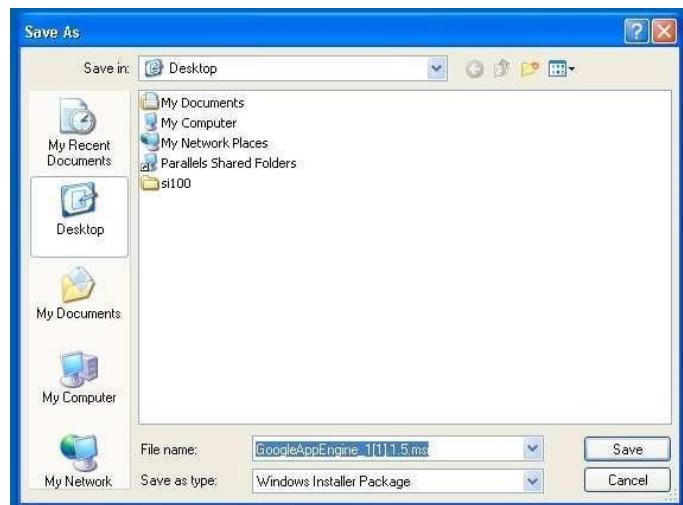
Platform	Version	Package	Size	SHA1 Checksum
Windows	1.1.5 - 10/03/08	GoogleAppEngine_1.1.5.msi	2.5 MB	e974312b4aefc0b3873ff0d93eb4c525d5e88c30
Mac OS X	1.1.5 - 10/03/08	GoogleAppEngineLauncher-1.1.5.dmg	3.6 MB	f62208ac01c1b3e39796e58100d5f1b2f052d3e7
Linux/Other Platforms	1.1.5 - 10/03/08	google_appengine_1.1.5.zip	2.6 MB	cbb9ce817bdabf1c4f181d9544864e55ee253de1

You can download the Google App Engine SDK by going to:

<http://code.google.com/appengine/downloads.html>

and download the appropriate install package.

Download the Windows installer—the simplest thing is to download it to your Desktop or another folder that you remember



Double Click on the **Google Application Engine** installer.



Click through the installation wizard, and it should install the App Engine. If you do not have Python 2.5, it will install Python 2.5 as well.

Once the install is complete you can discard the downloaded installer



RESULT:

DEPARTMENT OF CSE		
Preparation(Algorithm)	4	
Observation(Program)	4	
Results(Output)	4	
Interpretation(Validation)	4	
Viva-Voce	4	
Total	20	

EXP NO: 4

DATE:

INSTALL GOOGLE APP ENGINE –WEB APPLICATIONS

AIM

To use GAE launcher to launch the web applications

PROCEDURE

- Now you need to create a simple application.
- Use the “+” option to have the launcher make us an application– but instead we will do it by hand to get a better sense of what is going on.
- Make a folder for your Google App Engine applications. I am going to make a Folder on my Desktop called “**apps**” – the path to this folder is:

C:\Documents and Settings\csev\Desktop\apps

And then make a sub--folder in within apps called “ae--01--trivial” – the path to this folder would be:

C:\ Documents and Settings \csev\Desktop\apps\ae--01--trivial

Using a text editor such as JEdit (www.jedit.org), create a file called **app.yaml** in the **ae-01-trivial** folder with the following contents:

```
application: ae-01-
trivial version: 1
runtime: python
api_version: 1
```

handlers:

```
- url: /.*
  script: index.py
```

Note: Please do not copy and paste the se lines in to your text editor–you might end up with strange characters – simply type them into your editor.

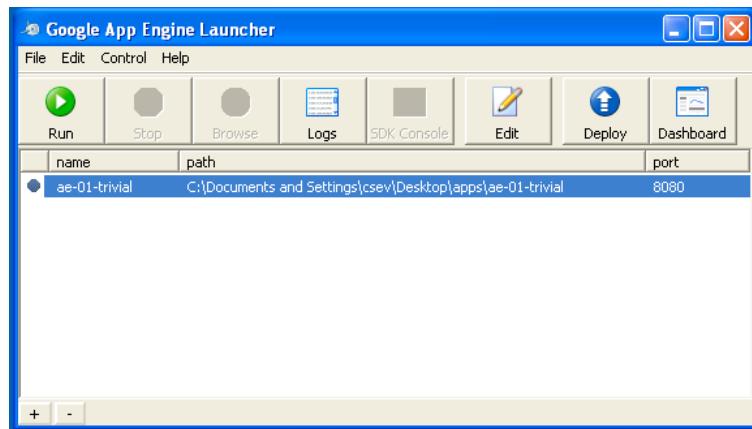
Then create a file in the **ae-01-trivial** folder called **index.py** with three lines in it:

```

print 'Content-Type:
text/plain' print ''
print 'Hello there Chuck'

```

Then start the **GoogleAppEngineLauncher** program that can be found under **Applications**. Use the **File > Add Existing Application** command and navigate into the **apps** directory and select the **ae-01-trivial** folder. Once you have added the application, select it so that you can control the application using the **launcher**.



Once you have selected your application and press **Run**. After a few moments your application will start and the launcher will show a little green icon next to your application. Then press **Browse** to open a browser pointing at your application which is running at **http://localhost:8080/**

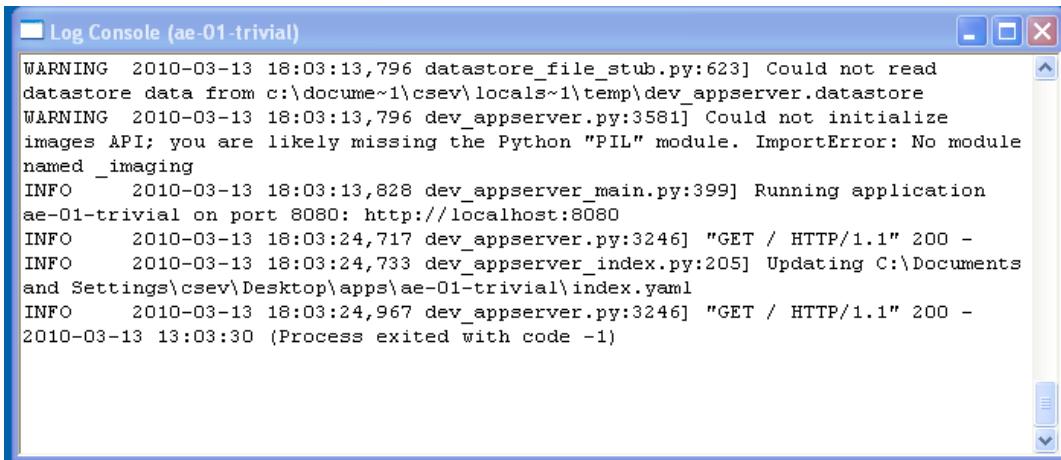
Paste **http://localhost:8080** into your browser and you should see your application as follows:



Just for fun, edit the **index.py** to change the name “Chuck” to your own name and press Refresh in the browser to verify your updates.

Watching the Log

You can watch the internal log of the action that the web server is performing when you are interacting with your application in the browser. Select your application in the Launcher and press the **Logs** button to bring up a log window:

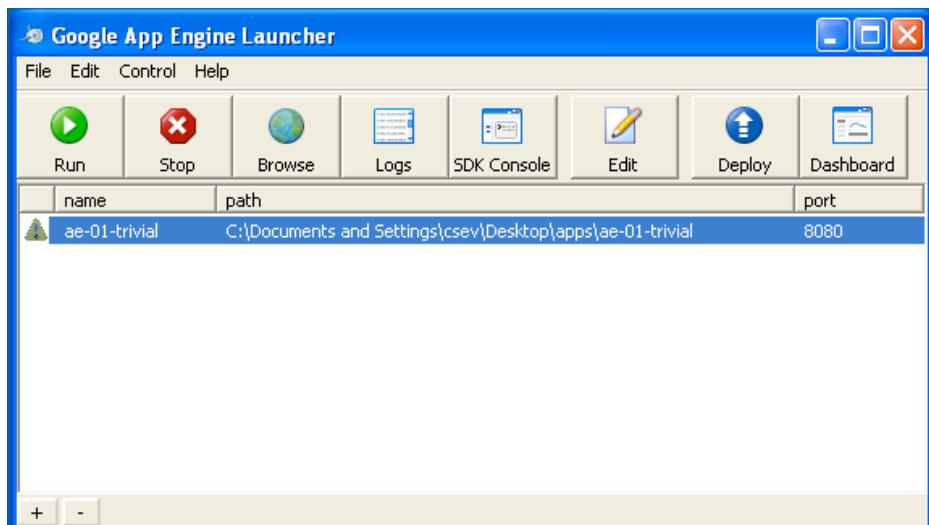


```
WARNING 2010-03-13 18:03:13,796 datastore_file_stub.py:623] Could not read
datastore data from c:\docume~1\csev\locals~1\temp\dev_appserver.datastore
WARNING 2010-03-13 18:03:13,796 dev_appserver.py:3581] Could not initialize
images API; you are likely missing the Python "PIL" module. ImportError: No module
named _imaging
INFO    2010-03-13 18:03:13,828 dev_appserver_main.py:399] Running application
ae-01-trivial on port 8080: http://localhost:8080
INFO    2010-03-13 18:03:24,717 dev_appserver.py:3246] "GET / HTTP/1.1" 200 -
INFO    2010-03-13 18:03:24,733 dev_appserver_index.py:205] Updating C:\Documents
and Settings\csev\Desktop\apps\ae-01-trivial\index.yaml
INFO    2010-03-13 18:03:24,967 dev_appserver.py:3246] "GET / HTTP/1.1" 200 -
2010-03-13 13:03:30 (Process exited with code -1)
```

Each time you press **Refresh** in your browser—you can see it retrieving the output with a **GET** request.

Dealing with Errors

With two files to edit, there are two general categories of errors that you may encounter. If you make a mistake on the **app.yaml** file, the AppEngine will not start and your launcher will show a yellow icon near your application:



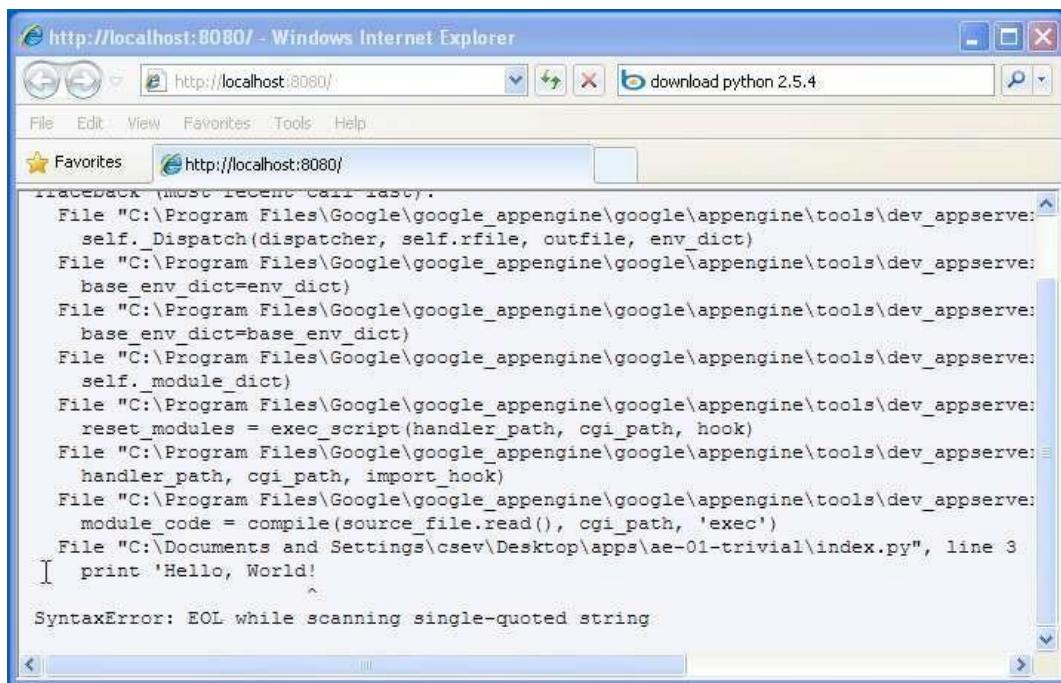
To get more detail on what is going wrong, take a look at the log for the application:

The screenshot shows a window titled "Log Console (ae-01-trivial)". The error message is:

```
invalid object:  
Unknown url handler type.  
<URLMap  
    static_dir=None  
    secure=default  
    script=None  
    url='.*'  
    static_files=None  
    upload=None  
    mime_type=None  
    login=optional  
    require_matching_file=None  
    auth_fail_action=redirect  
    expiration=None  
    >  
in "C:\Documents and Settings\csev\Desktop\apps\ae-01-trivial\app.yaml", line 8,  
column 1
```

In this instance – the mistake is mis-indenting the last line in the app.yaml (line 8).

If you make a syntax error in the index.py file, a Python trace back error will appear in your browser.



The error you need to see is likely to be the last few lines of the output – in this case I made a Python syntax error on line one of our one-line application.

Reference: http://en.wikipedia.org/wiki/Stack_trace

When you make a mistake in the **app.yaml** file – you must fix the mistake and attempt to start the application again. You make a mistake in a file like **index.py**, you can simply fix the file and press refresh in your browser – there is no need to restart the server.

Shutting Down the Server

To shut down the server, use the Launcher, select your application and press the Stop button.

This materials is Copyright All Rights Reserved– Charles Severance Comments and questions to csev@umich.edu www.dr--chuck.com

DEPARTMENT OF CSE		
Preparation(Algorithm)	4	
Observation(Program)	4	
Results(Output)	4	
Interpretation(Validation)	4	
Viva-Voce	4	
Total	20	

RESULT:

EXP NO: 5

DATE:

VIRUTAL MACHINES IN OPEN STACK

AIM:

To Create, Deploy and Launch Virtual Machines in OpenStack.

PROCEDURE:

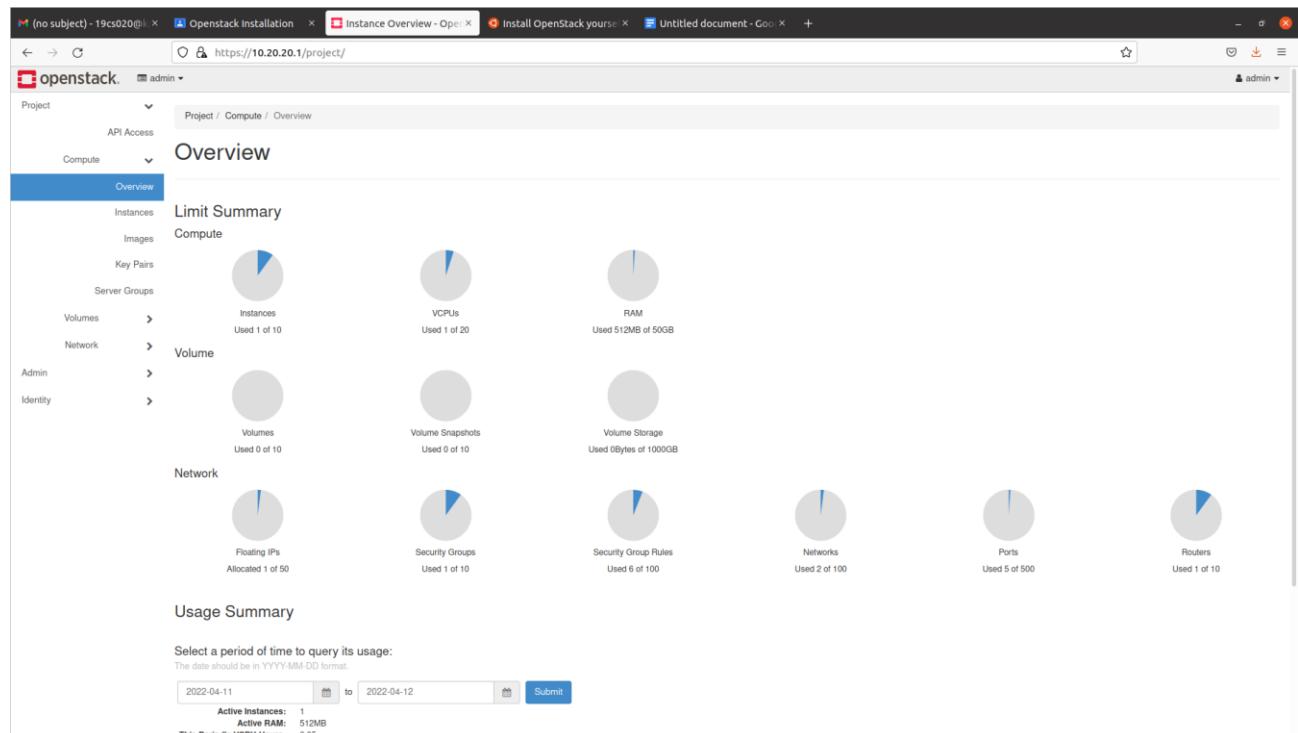
- Open The Terminal
- To install microstack beta with the following command
 - sudo snap install microstack --beta
- Information on the installed snap can be viewed like this:
 - snap list microstack
- The initialisation step automatically deploys, configures, and starts OpenStack services.
- In particular, it will create the database, networks, an image, several flavors, and ICMP/SSH security groups. This can all be done within 10 to 20 minutes depending on your machine
 - sudo microstack init --auto --control
- To list the default image:
 - microstack.openstack image list
- To get the default list of flavors:
 - microstack.openstack flavor list
- To create an instance named ‘test’ based on the ‘cirros’ image:
 - microstack launch cirros -n test
- To Access the cloud dashboard
You can open your browser with the IP address which was showing in the output of cirros test

User Name : Admin (Since Name of Machine)

Password: To get the password of the user name use the following code in terminal

- sudo snap get microstack config.credentials.keystone-password

OUTPUT:



OpenStack is an open-source software cloud computing platform. OpenStack is primarily used for deploying an infrastructure as a service (IaaS) solution like Amazon Web Service (AWS). In other words, you can *make your own AWS* by using OpenStack. If you want to try out OpenStack, **TryStack** is the easiest and free way to do it.

In order to try OpenStack in TryStack, you must register yourself by joining TryStack Facebook Group. The acceptance of group needs a couple days because it's approved manually. After you have been accepted in the TryStack Group, you can log in TryStack.

The Easiest Way To Try Out OpenStack. We've set up a large, growing cluster of hardware running OpenStack on x86. The best part? It's totally free for you to try & test your apps—thanks to our generous individual and corporate contributors.

For A Free Account:

[Join Our Facebook Group](#)

TryStack.org Homepage

I assume that you already join to the Facebook Group and login to the dashboard. After you log in to the TryStack, you will see the Compute Dashboard like:

Project

Compute

Overview

Instances

Volumes

Images

Access & Security

Network

Object Store

Identity

Limit Summary

- Instances Used 1 of 3
- VCPUs Used 1 of 5
- RAM Used 2,048 of 8,192
- Floating IPs Used 1 of 1
- Security Groups Used 1 of 10

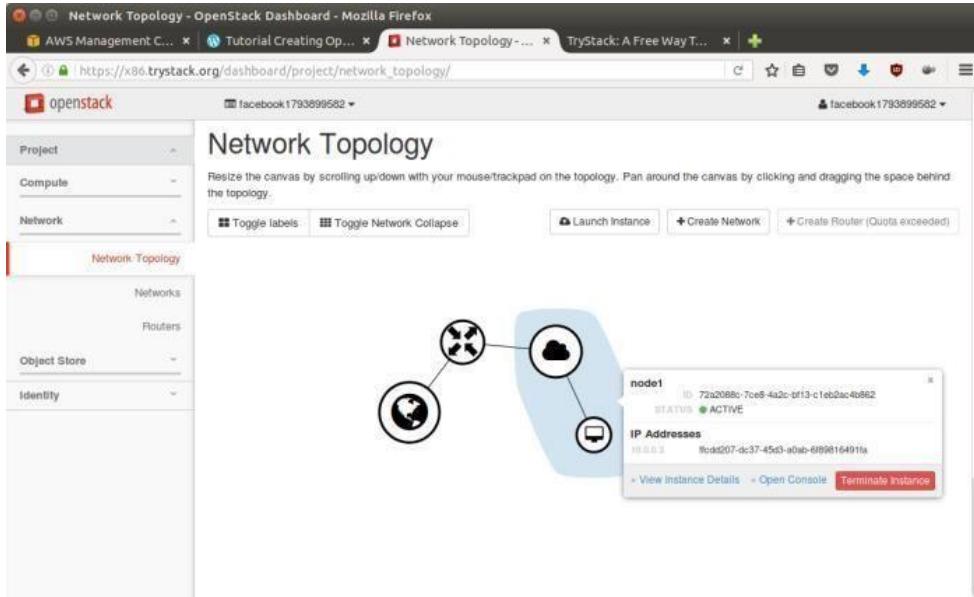
Usage Summary

Select a period of time to query its usage:

OpenStack Compute Dashboard

Overview: What we will do?

In this post, I will show you how to run an OpenStack instance. The instance will be accessible through the internet (have a public IP address). The final topology will like:



Network topology

As you see from the image above, the instance will be connected to a local network and the local network will be connected to internet.

Step 1: Create Network

Network? Yes, the network in here is our own local network. So, your instances will be not mixed up with the others. You can imagine this as your own LAN (Local Area Network) in the cloud.

- Go to **Network > Networks** and then click **Create Network**.
- In **Network** tab, fill **Network Name** for example internal and then click **Next**.
- In **Subnet** tab,
 - Fill **Network Address** with appropriate CIDR, for example 192.168.1.0/24. Use private network CIDR block as the best practice.
 - Select **IP Version** with appropriate IP version, in this case IPv4.
 - Click **Next**.
 - In **Subnet Details** tab, fill **DNS Name Servers** with 8.8.8.8 (Google DNS) and then click **Create**.

Step 2: Create Instance

Now, we will create an instance. The instance is a virtual machine in the cloud, like AWS EC2. You need the instance to connect to the network that we just created in the previous step.

- Go to **Compute > Instances** and then click **Launch Instance**.

- In **Details** tab,
 - Fill **Instance Name**, for example Ubuntu 1.
 - elect **Flavor**, for example m1.medium.
 - Fill **Instance Count** with 1.
 - Select **Instance Boot Source** with **Boot from Image**.
 - Select **Image Name** with **Ubuntu 14.04 amd64 (243.7 MB)** if you want install Ubuntu 14.04 in your virtual machine.
- In **Access & Security** tab,
 - Click [+] button of **Key Pair** to import key pair. This key pair is a public and private key that we will use to connect to the instance from our machine.
 - In **Import Key Pair** dialog,
 - Fill **Key Pair Name** with your machine name (for example Edward-Key).
 - Fill **Public Key** with your **SSH public key** (usually is in `~/.ssh/id_rsa.pub`). See description in Import Key Pair dialog box for more information. If you are using Windows, you can use **Puttygen** to generate key pair.
 - Click **Import key pair**.
 - In **Security Groups**, mark/check **default**.
 - In **Networking** tab,
 - In **Selected Networks**, select network that have been created in Step 1, for example `internal`.
 - Click **Launch**.
 - If you want to create multiple instances, you can repeat step 1-5. I created one more instance with instance name Ubuntu 2.

Step 3: Create Router

I guess you already know what router is. In the step 1, we created our network, but it is isolated. It doesn't connect to the internet. To make our network has an internet connection, we need a router that running as the gateway to the internet.

- Go to **Network > Routers** and then click **Create Router**.
- Fill **Router Name** for example `router1` and then click **Create router**.
- Click on your **router name link**, for example `router1`, **Router Details** page.
- Click **Set Gateway** button in upper right:

- Select **External networks** with **external**.
 - Then **OK**.
- Click **Add Interface** button.
 - Select **Subnet** with the network that you have been created in Step 1.
 - Click **Add interface**.
- Go to **Network > Network Topology**. You will see the network topology. In the example, there are two network, i.e. external and internal, those are bridged by a router. There are instances those are joined to internal network.

Step 4: Configure Floating IP Address

Floating IP address is public IP address. It makes your instance is accessible from the internet. When you launch your instance, the instance will have a private network IP, but no public IP. In OpenStack, the public IPs is collected in a pool and managed by admin (in our case is TryStack). You need to request a public (floating) IP address to be assigned to your instance.

- Go to **Compute > Instance**.
- In one of your instances, click **More > Associate Floating IP**.
- In **IP Address**, click Plus [+].
- Select **Pool to external** and then click **Allocate IP**.
- Click **Associate**.
- Now you will get a public IP, e.g. 8.21.28.120, for your instance.

Step 5: Configure Access & Security

OpenStack has a feature like a firewall. It can whitelist/blacklist your in/out connection. It is called *Security Group*.

- Go to **Compute > Access & Security** and then open **Security Groups** tab.
- In **default** row, click **Manage Rules**.
- Click **Add Rule**, choose **ALL ICMP** rule to enable ping into your instance, and then click **Add**.
- Click **Add Rule**, choose **HTTP** rule to open HTTP port (port 80), and then click **Add**.
- Click **Add Rule**, choose **SSH** rule to open SSH port (port 22), and then click **Add**.
- You can open other ports by creating new rules.

Step 6: SSH to Your Instance

Now, you can SSH your instances to the floating IP address that you got in the step 4. If you are using Ubuntu image, the SSH user will be ubuntu.

DEPARTMENT OF CSE		
Preparation(Algorithm)	4	
Observation(Program)	4	
Results(Output)	4	
Interpretation(Validation)	4	
Viva-Voce	4	
Total	20	

RESULT:

EXP NO: 6

DATE:

WAREHOUSE APPLICATION IN SALESFORCE

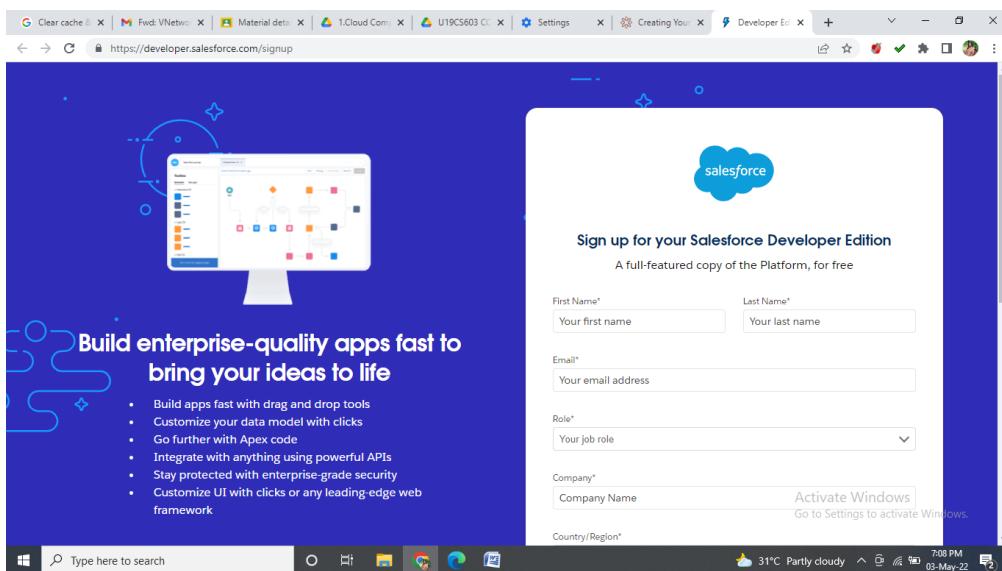
AIM:

To create a warehouse application in SaleForce.com

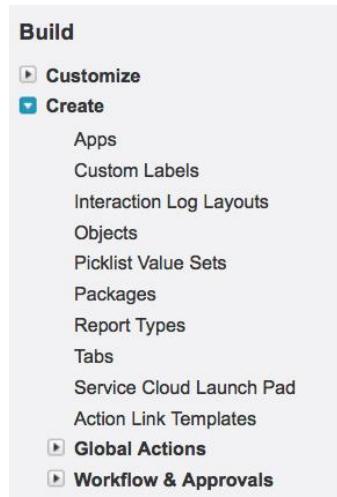
PROCEDURE:

- Register Salesforce.com developer account

<https://developer.salesforce.com/signup>



- Once registered, get valid username and password, then login to SalesForce.com
- Create simple Warehouse application



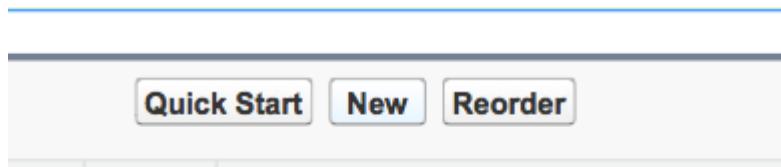
- To create an Application in Salesforce navigate to Setup | Build | Create | Apps.

The screenshot shows the Salesforce Setup interface. At the top, there is a search bar with 'Apps' typed in and a magnifying glass icon. Below the search bar are 'Expand All' and 'Collapse All' buttons. The main area is divided into sections: 'Administer' and 'Build'. Under 'Administer', there are two collapsed sections: 'Manage Apps' (which includes 'Connected Apps', 'Connected Apps OAuth Usage', and 'App Menu') and 'Google Apps' (which includes 'Google Apps Settings'). Under 'Build', there is a collapsed section 'Create' which contains an item 'Apps'. This 'Apps' item is highlighted with a red rectangular box.

Now click on Apps as show above. Now a group of apps will be displayed. The sitcom apps are checked in their respective check box and the remaining are standard Apps.

Apps					Quick Start	New	Reorder	Apps Help
Action	App Label	Console	Custom	Description				
Edit	App Launcher	<input type="checkbox"/>	<input type="checkbox"/>	App Launcher tabs				
Edit	Call Center	<input type="checkbox"/>	<input type="checkbox"/>	State-of-the-Art On-Demand Customer Service				
Edit	Community	<input type="checkbox"/>	<input type="checkbox"/>	Salesforce CRM Communities				
Edit	Content	<input type="checkbox"/>	<input type="checkbox"/>	Salesforce CRM Content				
Edit	Marketing	<input type="checkbox"/>	<input type="checkbox"/>	Best-in-class on-demand marketing automation				
Edit	Platform	<input type="checkbox"/>	<input type="checkbox"/>	The fundamental Force.com platform				
Edit	Sales	<input type="checkbox"/>	<input type="checkbox"/>	The world's most popular sales force automation (SFA) solution				
Edit	Salesforce Chatter	<input type="checkbox"/>	<input type="checkbox"/>	The Salesforce Chatter social network, including profiles and feeds				
Edit	Sample Console	<input checked="" type="checkbox"/>	<input type="checkbox"/>	The out-of-the box console for users who work with multiple records on one screen				
Edit	Site.com	<input type="checkbox"/>	<input type="checkbox"/>	Build pixel-perfect, data-rich websites using the drag-and-drop Site.com application, and manage content and published sites.				
Edit Del	Star Pharma	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Star Pharma Company application				

To create new custom application in Salesforce navigate to **Setup / Build / Create / Apps / New**. When we click on New button here we have to complete 5 steps to create an app in salesforce.



Click on New button as shown above.

New Custom App

[Help for this Page](#) ?

Step 1. Select Type Step 1 of 5

Select the type of app to create.

Custom app
 Console

[Next](#) [Cancel](#)

Now select Custom app and click on Next Button.

Step 2. Enter the Details Step 2 of 5

Fill in the fields below to define the custom app.

Custom App Information | = Required Information

App Label	<input type="text" value="Tutorial kart"/>	Example: HRforce, Financeforce, Bugforce
App Name	<input type="text" value="Tutorial_kart"/> i	
Description	<input type="text"/>	

[Previous](#) [Next](#) [Cancel](#)

Enter App Label and APP Name and Description and finally click on next button.

Step 3. Choose the Image Source for the Custom App Logo

Step 3 of 5

Optionally, specify a logo for this custom app. To do so, choose an image file from the document library.



The file size of a custom app logo must be smaller than 20 KB. (For comparison, the salesforce.com logo is about 3 KB). To upload an image file, add a new document to the Documents tab. Image dimensions should be a maximum of 300 pixels wide by 55 pixels high for best results. Larger images will be resized and may appear distorted.

Insert an Image



[Previous](#) [Next](#) [Cancel](#)

If you want to choose an Image Insert an Image and click on Next Button.

Step 4. Choose the Tabs

Step 4 of 5

Choose the tabs to include in this custom app.

Available Tabs

- Price Books
- Accounts
- Campaigns
- Cases
- Contacts
- Contracts
- Dashboards
- Documents
- Forecasts
- Leads
- Opportunities
- Orders
- Products
- Reports

Add

Remove

Selected Tabs

Home

Up
Down

Default Landing Tab

Home



[Previous](#) [Next](#) [Cancel](#)

Choose the list of tabs to include in this Custom Application and click on Next button.

Finally choose the list of profile to which this custom app is made to visible and finally click on Save button.

Apps					Quick Start	New	Reorder	Apps Help 
Action	App Label	Console	Custom	Description				
Edit	App Launcher	<input type="checkbox"/>	<input type="checkbox"/>	App Launcher tabs				
Edit	Call Center	<input type="checkbox"/>	<input type="checkbox"/>	State-of-the-Art On-Demand Customer Service				
Edit	Community	<input type="checkbox"/>	<input type="checkbox"/>	Salesforce CRM Communities				
Edit	Content	<input type="checkbox"/>	<input type="checkbox"/>	Salesforce CRM Content				
Edit	Marketing	<input type="checkbox"/>	<input type="checkbox"/>	Best-in-class on-demand marketing automation				
Edit	Platform	<input type="checkbox"/>	<input type="checkbox"/>	The fundamental Force.com platform				
Edit	Sales	<input type="checkbox"/>	<input type="checkbox"/>	The world's most popular sales force automation (SFA) solution				
Edit	Salesforce Chatter	<input type="checkbox"/>	<input type="checkbox"/>	The Salesforce Chatter social network, including profiles and feeds				
Edit	Sample Console	<input checked="" type="checkbox"/>	<input type="checkbox"/>	The out-of-the box console for users who work with multiple records on one screen				
Edit	Site.com	<input type="checkbox"/>	<input type="checkbox"/>	Build pixel-perfect, data-rich websites using the drag-and-drop Site.com application, and manage content and published sites.				
Edit Del	Star Pharma	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Star Pharma Company application				
Edit Del	Tutorial kart	<input type="checkbox"/>	<input checked="" type="checkbox"/>					
Edit Del	Tutorials kart	<input type="checkbox"/>	<input checked="" type="checkbox"/>					

DEPARTMENT OF CSE		
Preparation(Algorithm)	4	
Observation(Program)	4	
Results(Output)	4	
Interpretation(Validation)	4	
Viva-Voce	4	
Total	20	

RESULT:

EXP NO: 7 INSTALL HADOOP AND RUN APPLICATIONS (WORD COUNT)

DATE:

AIM:

To install hadoop and run wordcount program in bigdata environment.

Procedure:

Step 1: Download the Java 8 Package. Save this file in your home directory.

Step 2: Extract the Java Tar File.

Command: `tar -xvf jdk-8u101-linux-i586.tar.gz`

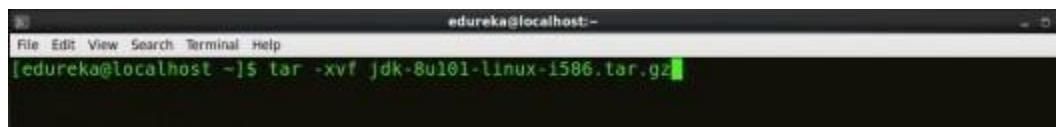


Fig: Hadoop Installation – Extracting Java Files

Step 3: Download the Hadoop 2.7.3 Package.

Command: `wget`

<https://archive.apache.org/dist/hadoop/core/hadoop-2.7.3/hadoop-2.7.3.tar.gz>

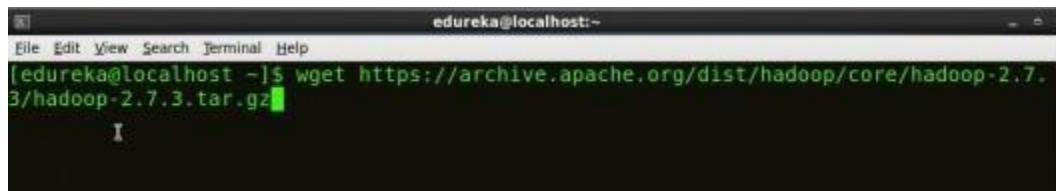


Fig: Hadoop Installation – Downloading Hadoop

Step 4: Extract the Hadoop tar File.

Command: `tar -xvf hadoop-2.7.3.tar.gz`

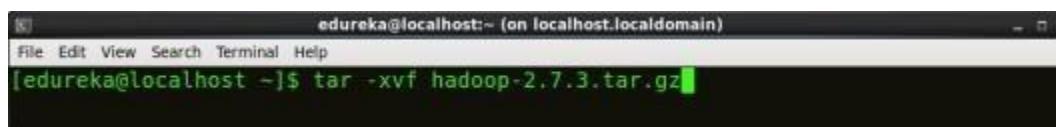
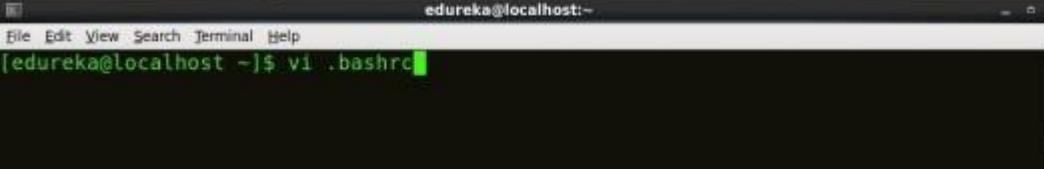


Fig: Hadoop Installation – Extracting Hadoop Files

Step 5: Add the Hadoop and Java paths in the bash file

(.bashrc). Open **.bashrc** file. Now, add Hadoop and Java Path as shown below.

Command: vi .bashrc



```
# User specific aliases and functions

export HADOOP_HOME=$HOME/hadoop-2.7.3
export HADOOP_CONF_DIR=$HOME/hadoop-2.7.3/etc/hadoop
export HADOOP_MAPRED_HOME=$HOME/hadoop-2.7.3
export HADOOP_COMMON_HOME=$HOME/hadoop-2.7.3
export HADOOP_HDFS_HOME=$HOME/hadoop-2.7.3
export YARN_HOME=$HOME/hadoop-2.7.3
export PATH=$PATH:$HOME/hadoop-2.7.3/bin

# Set JAVA_HOME

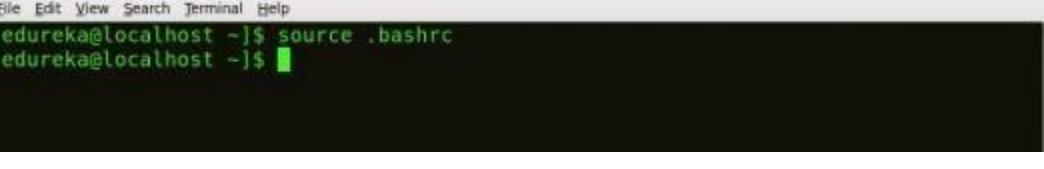
export JAVA_HOME=/home/edureka/jdk1.8.0_101
export PATH=/home/edureka/jdk1.8.0_101/bin:$PATH
```

Fig: Hadoop Installation – Setting Environment Variable

Then, save the bash file and close it.

For applying all these changes to the current Terminal, execute the source command.

Command: source .bashrc

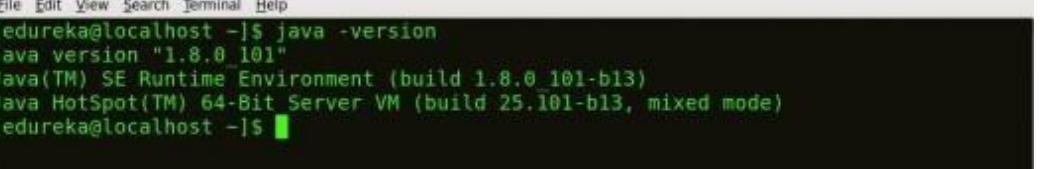


```
[edureka@localhost ~]$ source .bashrc
[edureka@localhost ~]$
```

Fig: Hadoop Installation – Refreshing environment variables

To make sure that Java and Hadoop have been properly installed on your system and can be accessed through the Terminal, execute the java -version and hadoop version commands.

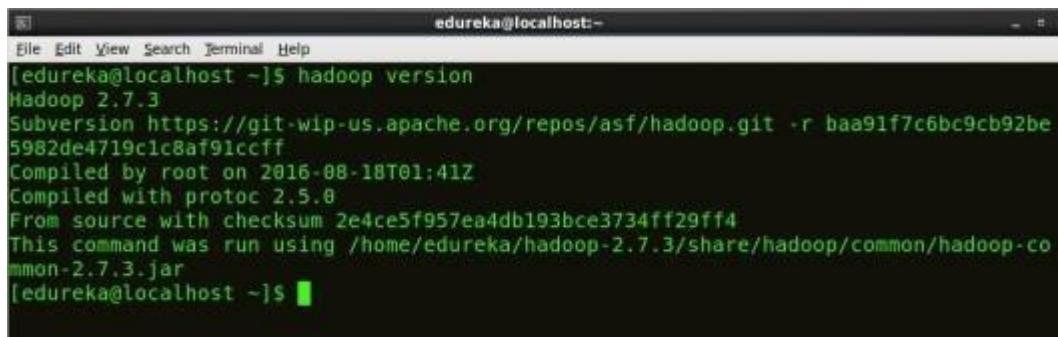
Command: java -version



```
[edureka@localhost ~]$ java -version
java version "1.8.0_101"
Java(TM) SE Runtime Environment (build 1.8.0_101-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.101-b13, mixed mode)
[edureka@localhost ~]$
```

Fig: Hadoop Installation – Checking Java Version

Command: hadoop version



```
[edureka@localhost ~]$ hadoop version
Hadoop 2.7.3
Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git -r baa91f7c6bc9cb92be
5982de4719c1c8af91ccff
Compiled by root on 2016-08-18T01:41Z
Compiled with protoc 2.5.0
From source with checksum 2e4ce5f957ea4db193bce3734ff29ff4
This command was run using /home/edureka/hadoop-2.7.3/share/hadoop/common/hadoop-common-2.7.3.jar
[edureka@localhost ~]$
```

Fig: Hadoop Installation – Checking Hadoop Version

Step 6: Edit the Hadoop Configuration files.

Command: cd hadoop-2.7.3/etc/hadoop/

Command: ls

All the Hadoop configuration files are located in **hadoop-2.7.3/etc/hadoop** directory as you can see in the snapshot below:



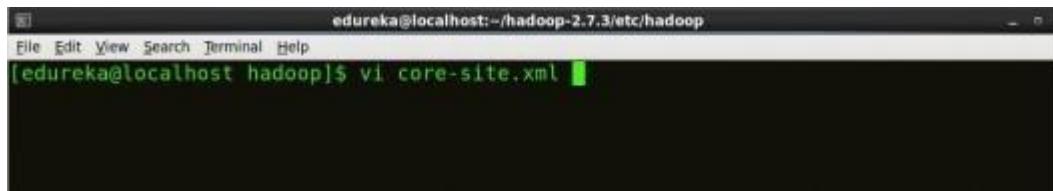
```
[edureka@localhost ~]$ cd hadoop-2.7.3/etc/hadoop/
[edureka@localhost hadoop]$ ls
capacity-scheduler.xml      httpfs-env.sh          mapred-env.sh
configuration.xml           httpfs-log4j.properties   mapred-queues.xml.template
container-executor.cfg       httpfs-signature.secret  mapred-site.xml.template
core-site.xml                httpfs-site.xml        slaves
hadoop-env.cmd              kms-acls.xml          ssl-client.xml.example
hadoop-env.sh                kms-env.sh            ssl-server.xml.example
hadoop-metrics2.properties  kms-log4j.properties   yarn-env.cmd
hadoop-metrics.properties   kms-site.xml          yarn-env.sh
hadoop-policy.xml            log4j.properties     yarn-site.xml
hdfs-site.xml                mapred-env.cmd
```

Fig: Hadoop Installation – Hadoop Configuration Files

Step 7: Open *core-site.xml* and edit the property mentioned below inside configuration tag:

core-site.xml informs Hadoop daemon where NameNode runs in the cluster. It contains configuration settings of Hadoop core such as I/O settings that are common to HDFS & MapReduce.

Command: vi core-site.xml



```
edureka@localhost:~/hadoop-2.7.3/etc/hadoop$ vi core-site.xml
```

```
<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:9000</value>
</property>
</configuration>
```

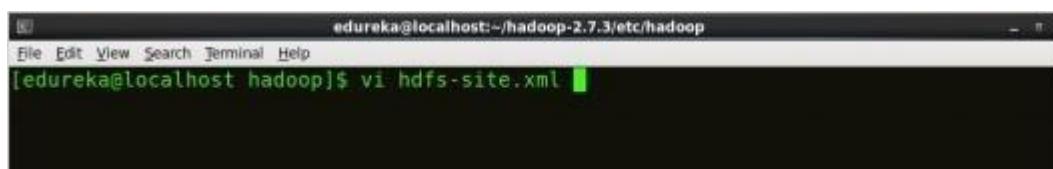
Fig: Hadoop Installation – Configuring core-site.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <?xmlstylesheet type="text/xsl" href="configuration.xsl"?>
3 <configuration>
4   <property>
5     <name>fs.default.name</name>
6     <value>hdfs://localhost:9000</value>
7   </property>
8 </configuration>
```

Step 8: Edit *hdfs-site.xml* and edit the property mentioned below inside configuration tag:

hdfs-site.xml contains configuration settings of HDFS daemons (i.e. NameNode, DataNode, Secondary NameNode). It also includes the replication factor and block size of HDFS.

Command: vi hdfs-site.xml



```
edureka@localhost:~/hadoop-2.7.3/etc/hadoop$ vi hdfs-site.xml
```

```
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.permission</name>
<value>false</value>
</property>
```

Fig: Hadoop Installation – Configuring hdfs-site.xml

```

1
2           <?xml version="1.0" encoding="UTF-8"?>
3   <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
4       <configuration>
5           <property>
6               <name>dfs.replication</name>
7               <value>1</value>
8           </property>
9           <property>
10              <name>dfs.permission</name>
11              <value>false</value>
12          </property>
13      </configuration>

```

Step 9: Edit the *mapred-site.xml* file and edit the property mentioned below inside configuration tag:

mapred-site.xml contains configuration settings of MapReduce application like number of JVM that can run in parallel, the size of the mapper and the reducer process, CPU cores available for a process, etc.

In some cases, *mapred-site.xml* file is not available. So, we have to create the *mapred-site.xml* file using *mapred-site.xml* template.

Command: cp *mapred-site.xml.template* *mapred-site.xml*

Command: vi *mapred-site.xml*.

```
edureka@localhost:~/hadoop-2.7.3/etc/hadoop
[edureka@localhost hadoop]$ cp mapred-site.xml.template mapred-site.xml
[edureka@localhost hadoop]$
```

```
edureka@localhost:~/hadoop-2.7.3/etc/hadoop
[edureka@localhost hadoop]$ vi mapred-site.xml
```

```

<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
</configuration>

```

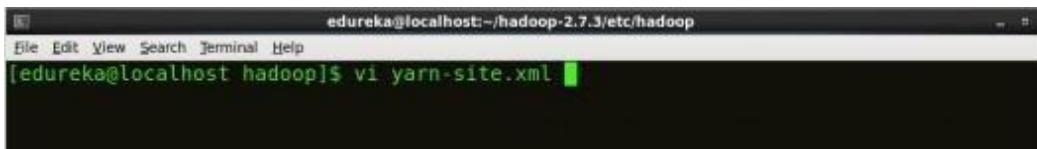
Fig: Hadoop Installation – Configuring mapred-site.xml

```
1           <?xml version="1.0" encoding="UTF-8"?>
2   <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
3       <configuration>
4           <property>
5               <name>mapreduce.framework.name</name>
6               <value>yarn</value>
7           </property>
8       </configuration>
```

Step 10: Edit *yarn-site.xml* and edit the property mentioned below inside configuration tag:

yarn-site.xml contains configuration settings of ResourceManager and NodeManager like application memory management size, the operation needed on program & algorithm, etc.

Command: vi *yarn-site.xml*



```
<configuration>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
</configuration>
```

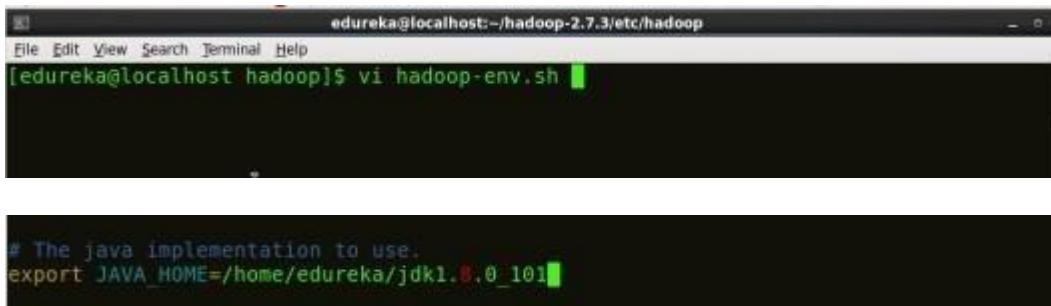
Fig: Hadoop Installation – Configuring *yarn-site.xml*

```
1
2           <?xml version="1.0">
3               <configuration>
4                   <property>
5                       <name>yarn.nodemanager.aux-services</name>
6                       <value>mapreduce_shuffle</value>
7                   </property>
8                   <property>
9                       <name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</
9                           name>
10                      <value>org.apache.hadoop.mapred.ShuffleHandler</value>
11                  </property>
12              </configuration>
```

Step 11: Edit *hadoop-env.sh* and add the Java Path as mentioned below:

hadoop-env.sh contains the environment variables that are used in the script to run Hadoop like Java home path, etc.

Command: vi *hadoop-env.sh*



```
# The java implementation to use.
export JAVA_HOME=/home/edureka/jdk1.8.0_101
```

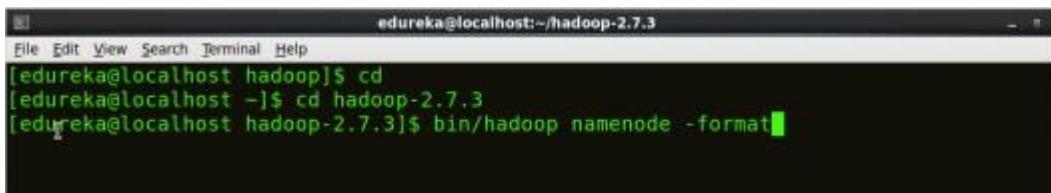
Fig: Hadoop Installation – Configuring *hadoop-env.sh*

Step 12: Go to Hadoop home directory and format the NameNode.

Command: cd

Command: cd *hadoop-2.7.3*

Command: bin/hadoop namenode -format



```
[edureka@localhost ~]$ cd
[edureka@localhost ~]$ cd hadoop-2.7.3
[edureka@localhost hadoop-2.7.3]$ bin/hadoop namenode -format
```

Fig: Hadoop Installation – Formatting NameNode

This formats the HDFS via NameNode. This command is only executed for the first time. Formatting the file system means initializing the directory specified by the *dfs.name.dir* variable.

Never format, up and running Hadoop filesystem. You will lose all your data stored in theHDFS.

Step 13: Once the NameNode is formatted, go to *hadoop-2.7.3/sbin* directory and start allthe daemons.

Command: cd hadoop-2.7.3/sbin
Either you can start all daemons with a single command or do it individually.

Command: ./start-all.sh

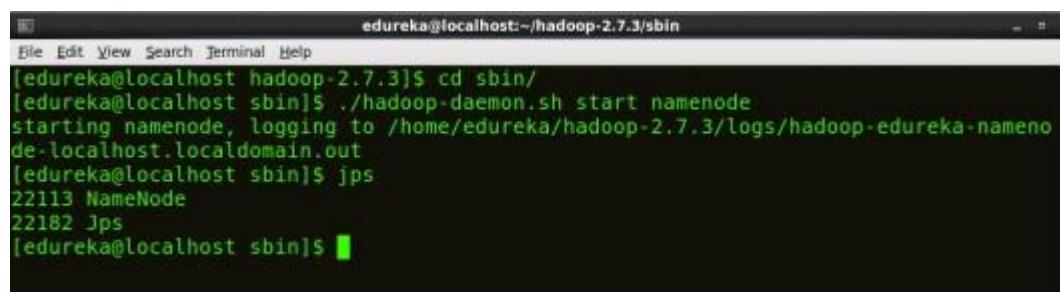
The above command is a combination of *start-dfs.sh*, *start-yarn.sh* & *mr-jobhistory-daemon.sh*

Or you can run all the services individually as below:

Start NameNode:

The NameNode is the centerpiece of an HDFS file system. It keeps the directory tree of all files stored in the HDFS and tracks all the file stored across the cluster.

Command: ./hadoop-daemon.sh start namenode



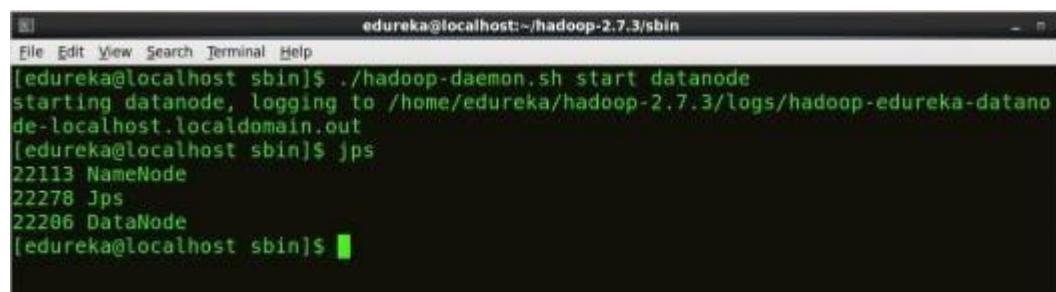
```
edureka@localhost:~/hadoop-2.7.3/sbin
File Edit View Search Terminal Help
[edureka@localhost hadoop-2.7.3]$ cd sbin/
[edureka@localhost sbin]$ ./hadoop-daemon.sh start namenode
starting namenode, logging to /home/edureka/hadoop-2.7.3/logs/hadoop-edureka-namenode-localhost.localdomain.out
[edureka@localhost sbin]$ jps
22113 NameNode
22182 Jps
[edureka@localhost sbin]$
```

Fig: Hadoop Installation – Starting NameNode

Start DataNode:

On startup, a DataNode connects to the Namenode and it responds to the requests from the Namenode for different operations.

Command: ./hadoop-daemon.sh start datanode



```
edureka@localhost:~/hadoop-2.7.3/sbin
File Edit View Search Terminal Help
[edureka@localhost sbin]$ ./hadoop-daemon.sh start datanode
starting datanode, logging to /home/edureka/hadoop-2.7.3/logs/hadoop-edureka-datanode-localhost.localdomain.out
[edureka@localhost sbin]$ jps
22113 NameNode
22278 Jps
22286 DataNode
[edureka@localhost sbin]$
```

Fig: Hadoop Installation – Starting DataNode

Start ResourceManager:

ResourceManager is the master that arbitrates all the available cluster resources and thus helps in managing the distributed applications running on the YARN system. Its work is to manage each NodeManagers and the each application's ApplicationMaster.

Command: ./yarn-daemon.sh start resourcemanager

```

edureka@localhost sbin$ ./yarn-daemon.sh start resourcemanager
starting resourcemanager, logging to /home/edureka/hadoop-2.7.3/logs/yarn-edureka-resourcemanager-localhost.localdomain.out
[edureka@localhost sbin]$ jps
22113 NameNode
22310 ResourceManager
22345 Jps
22206 DataNode
[edureka@localhost sbin]$

```

Fig: Hadoop Installation – Starting ResourceManager

Start NodeManager:

The NodeManager in each machine framework is the agent which is responsible for managing containers, monitoring their resource usage and reporting the same to the ResourceManager.

Command: ./yarn-daemon.sh start nodemanager

```

edureka@localhost sbin$ ./yarn-daemon.sh start nodemanager
starting nodemanager, logging to /home/edureka/hadoop-2.7.3/logs/yarn-edureka-nodemanager-localhost.localdomain.out
[edureka@localhost sbin]$ jps
22592 Jps
22113 NameNode
22310 ResourceManager
22206 DataNode
22559 NodeManager
[edureka@localhost sbin]$

```

Fig: Hadoop Installation – Starting NodeManager

Start JobHistoryServer:

Job History Server is responsible for servicing all job history related requests from client.

Command: ./mr-jobhistory-daemon.sh start historyserver

Step 14: To check that all the Hadoop services are up and running, run the below command.

Command: jps

```

edureka@localhost sbin$ ./mr-jobhistory-daemon.sh start historyserver
starting historyserver, logging to /home/edureka/hadoop-2.7.3/logs/mapred-edureka-historyserver-localhost.localdomain.out
[edureka@localhost sbin]$ jps
22113 NameNode
22310 ResourceManager
22694 JobHistoryServer
22727 Jps
22206 DataNode
22559 NodeManager
[edureka@localhost sbin]$

```

Fig: Hadoop Installation – Checking Daemons

Step 15: Now open the Mozilla browser and go to **localhost:50070/dfshealth.html** to check the NameNode interface.

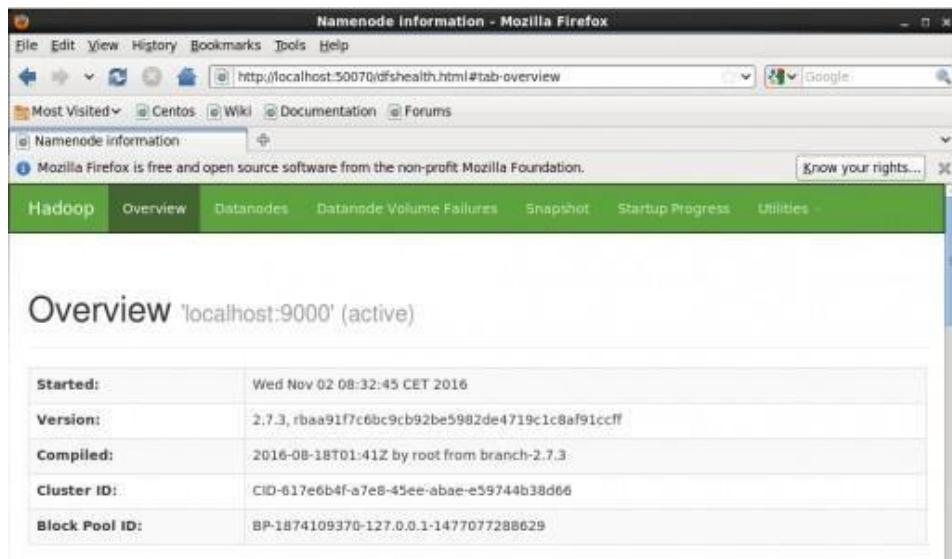


Fig: Hadoop Installation – Starting WebUI

Congratulations, you have successfully installed a single node Hadoop cluster

DEPARTMENT OF CSE		
Preparation(Algorithm)	4	
Observation(Program)	4	
Results(Output)	4	
Interpretation(Validation)	4	
Viva-Voce	4	
Total	20	

RESULT:

EXP NO: 8

AWS – CREATE AND LAUNCH VM IN EC2

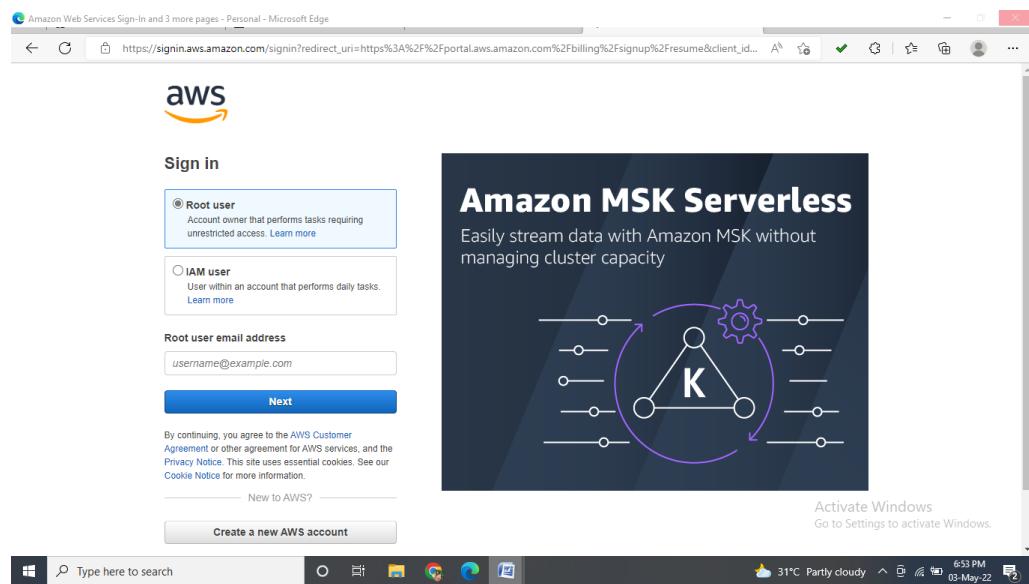
DATE:

AIM:

To sign up for a free account in AWS, create and launch your VM in the EC2 Cloud and perform web hosting in the VM.

PROCEDURE:

- Sign up – free account in aws



- Before creating an instance in AWS, create an account in AWS using your debit card details and then Log in to your account.
- Search EC2 and click “Launch Instance” then the steps to launch instance will be followed.

- In step 1, check the Free Tier check box and select Windows server 2016 base64bit which is eligible for free tier.

- Step 2, choose instance type which is eligible for free tier(General purpose, t2micro) and click Next.

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more about instance types and how they can meet your computing needs.](#)

Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
General purpose	t2.micro <small>Free tier eligible</small>	1	1	EBS only	-	Low to Moderate	Yes
General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes

Review and Launch | **Next: Configure Instance Details**

- In step 3, Enter configure instance details where you should select Number of instances as 1 and click Next.

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of Instances	<input type="text" value="1"/>	Launch into Auto Scaling Group
Purchasing option	<input type="checkbox"/> Request Spot Instances	
Network	<input type="text" value="vpc-be8314d6 (default)"/>	<input type="button" value="Create new VPC"/>
Subnet	<input type="text" value="No preference (default subnet in any Availability Zone)"/>	<input type="button" value="Create new subnet"/>
Auto-assign Public IP	<input type="checkbox"/> Use subnet setting (Enable)	
Domain join directory	<input type="text" value="None"/>	<input type="button" value="Create new directory"/>
IAM role	<input type="text" value="None"/>	<input type="button" value="Create new IAM role"/>
Shutdown behavior	<input type="text" value="Stop"/>	
Enable termination protection	<input type="checkbox"/> Protect against accidental termination	
Monitoring	<input type="checkbox"/> Enable CloudWatch detailed monitoring	

Review and Launch | **Next: Add Storage**

- Step 4, choose storage where they can offer 30 GB of storage which is eligible for free tier users and click Next.

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more about storage options in Amazon EC2.](#)

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/sda1	snap-06b1d7ead7bb73f64	30	General Purpose SSD (GP2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more about free usage tier eligibility and usage restrictions.](#)

Cancel Previous Review and Launch Next: Add Tags

- In step 5, Add tags by clicking Add tag by entering Key and value and clickNext.

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more about tagging your Amazon EC2 resources.](#)

Key	(127 characters maximum)	Value	(255 characters maximum)	Instances	Volumes
name		sample		<input checked="" type="checkbox"/>	<input type="checkbox"/>

Add another tag (Up to 50 tags maximum)

Cancel Previous Review and Launch Next: Configure Security Group

Step 6, configure security group by clicking Add rule where the source should be given as My IP and click Review and Launch.

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more about Amazon EC2 security groups.](#)

Assign a security group: Create a new security group Select an existing security group

Security group name:

Description:

Type	Protocol	Port Range	Source	Description
Custom TCP	TCP	0	My IP	e.g. SSH for Admin Desktop

Add Rule

Warning
You will not be able to connect to this instance as the AMI requires port(s) 3389 to be open in order to have access. Your current security group doesn't have port(s) 3389 open.

Buttons: Cancel, Previous, **Review and Launch**

- i. In step 7, Check all the details entered and click Launch.

Step 7: Review Instance Launch

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups [Edit security groups](#)

Security group name: launch-wizard-1
Description: launch-wizard-1 created 2018-04-03T03:27:50.560+05:30

Type	Protocol	Port Range	Source	Description
Custom TCP Rule	TCP	0	117.239.104.11/32	

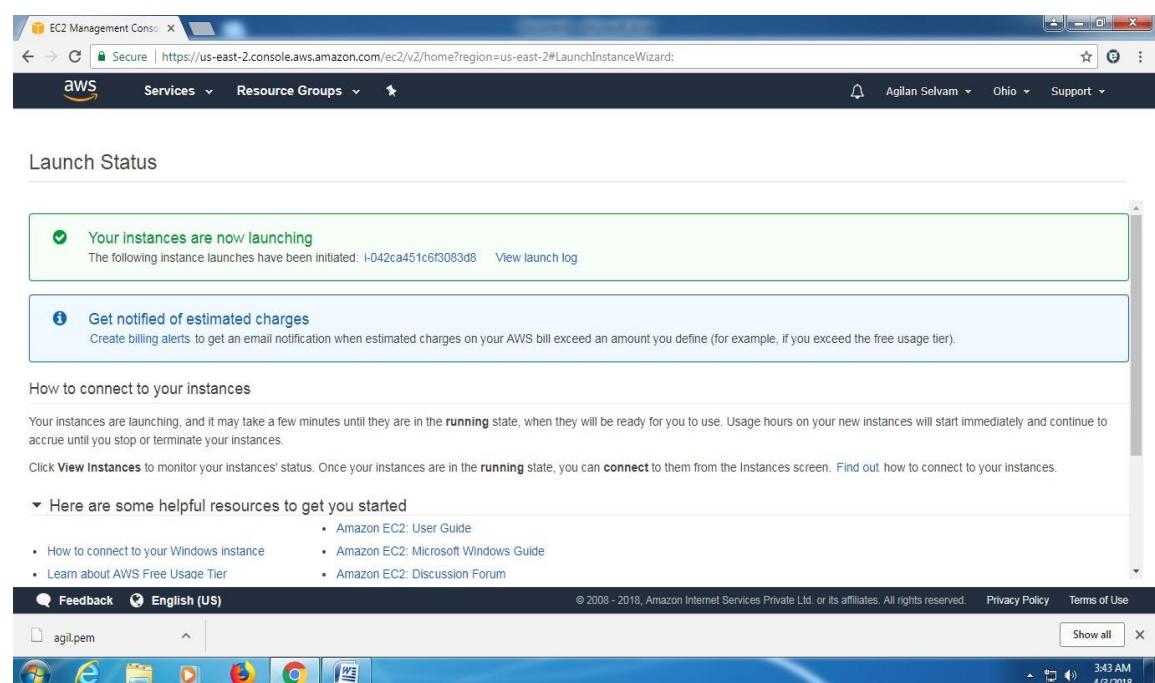
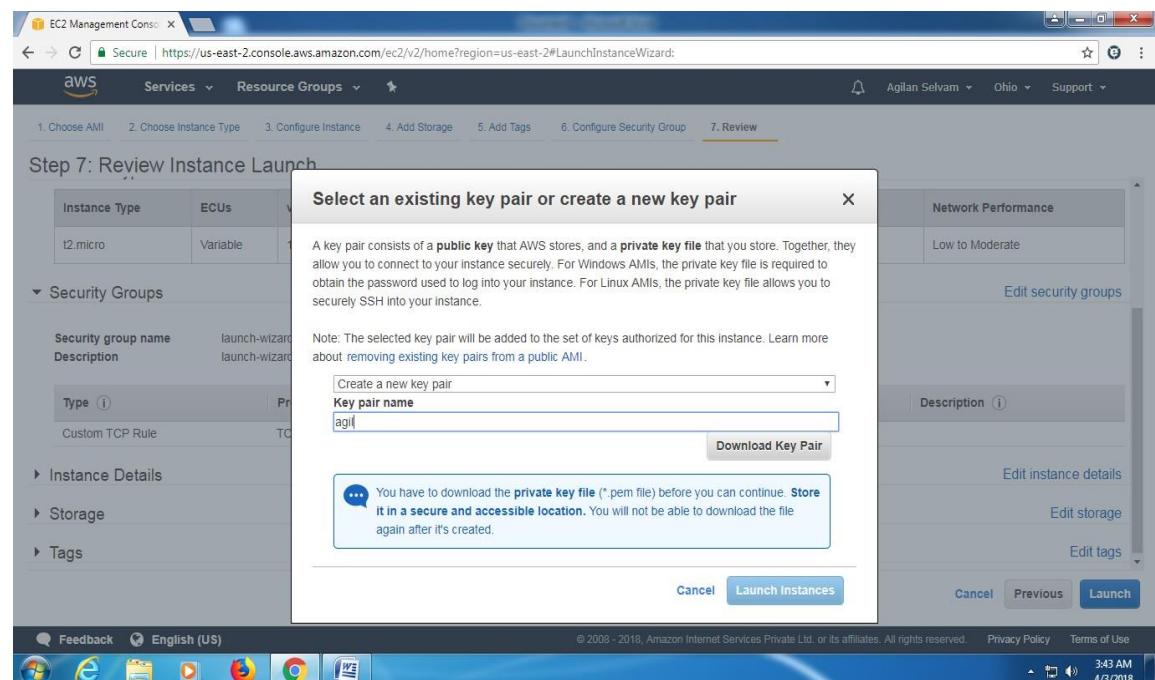
Instance Details [Edit instance details](#)

Storage [Edit storage](#)

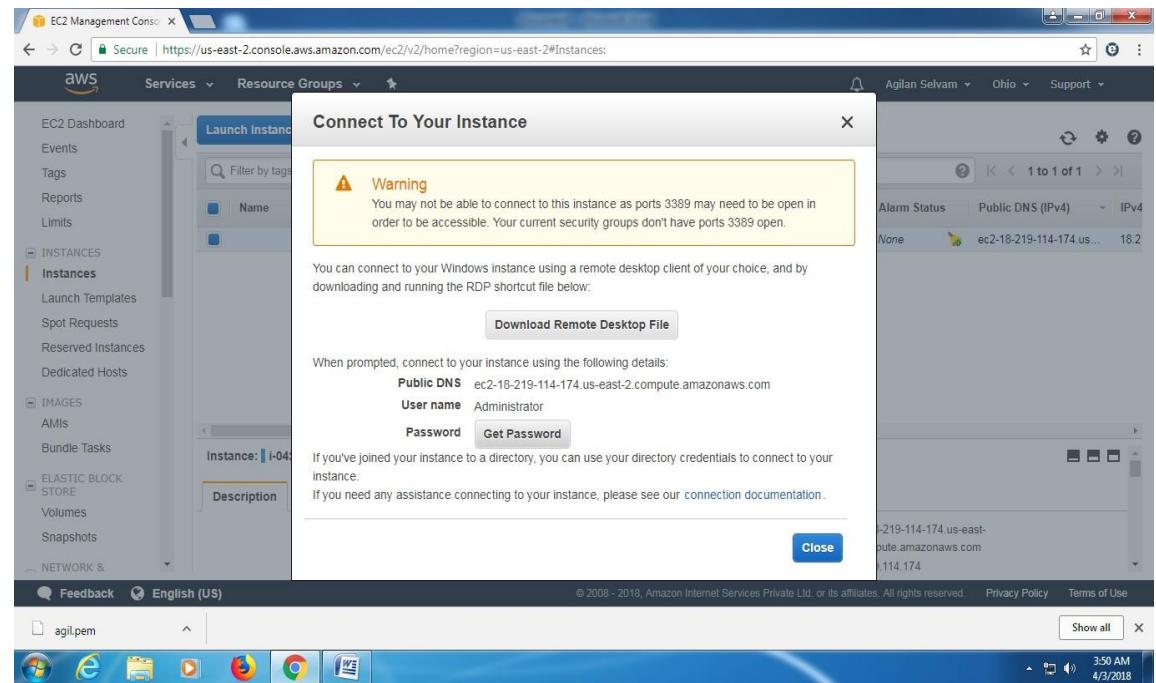
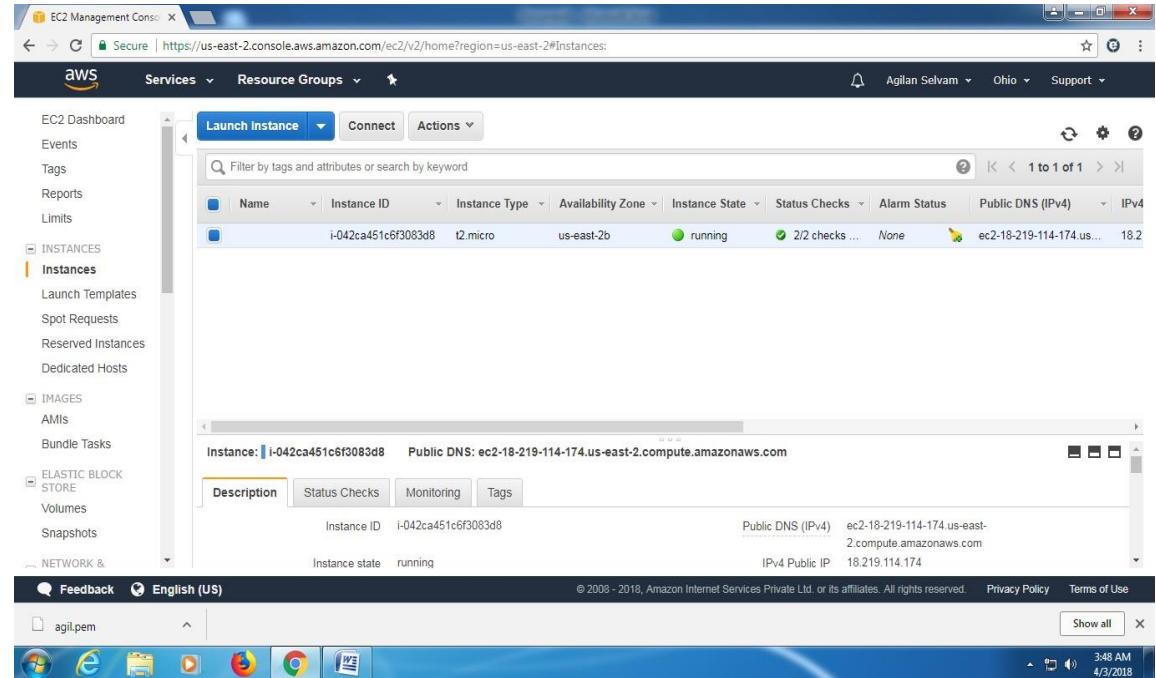
Tags [Edit tags](#)

Buttons: Cancel, Previous, **Launch**

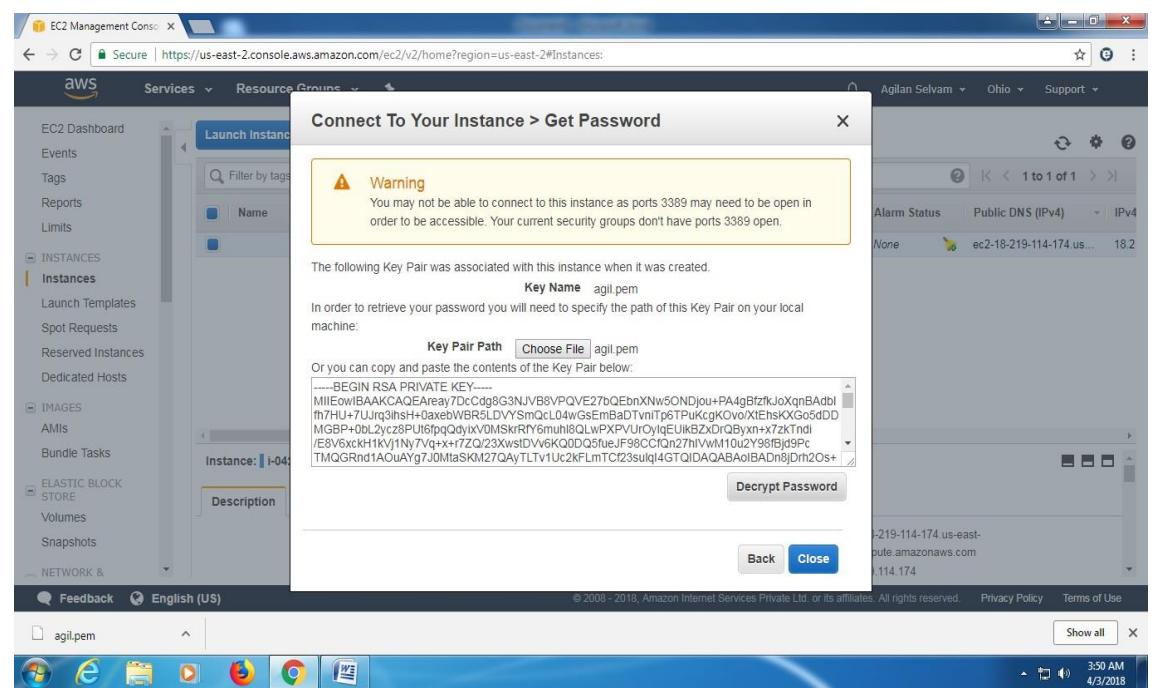
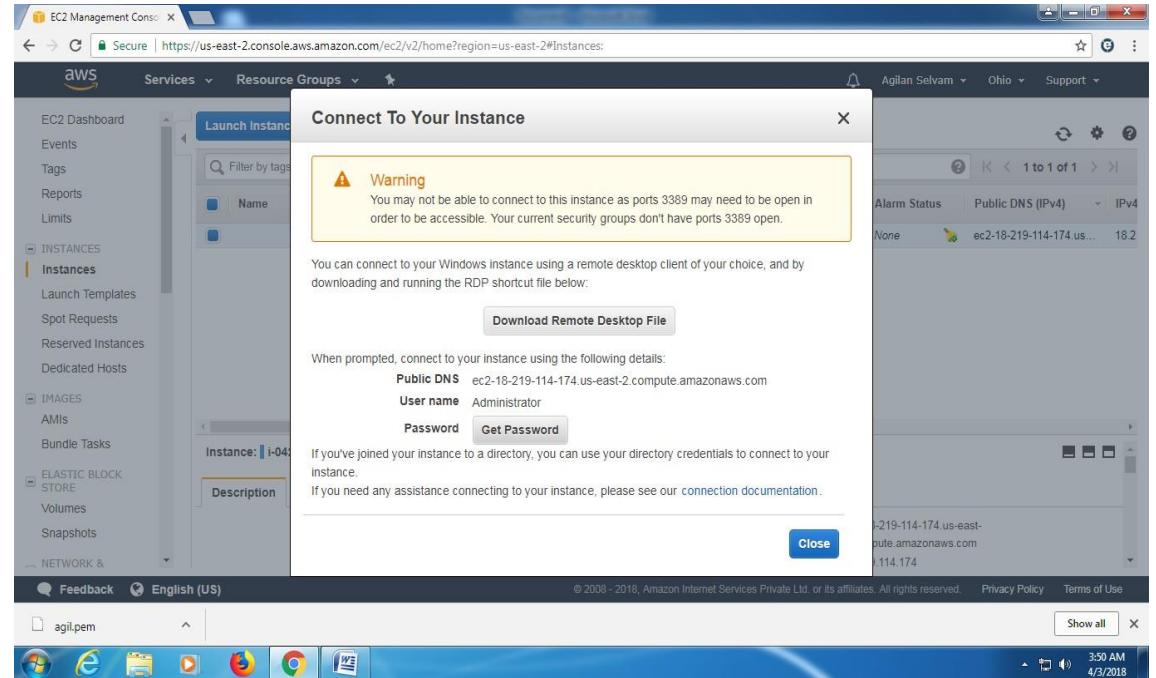
After clicking Launch, enter the key pair name and click Download key pair and save the file locally and click Launch Instance.

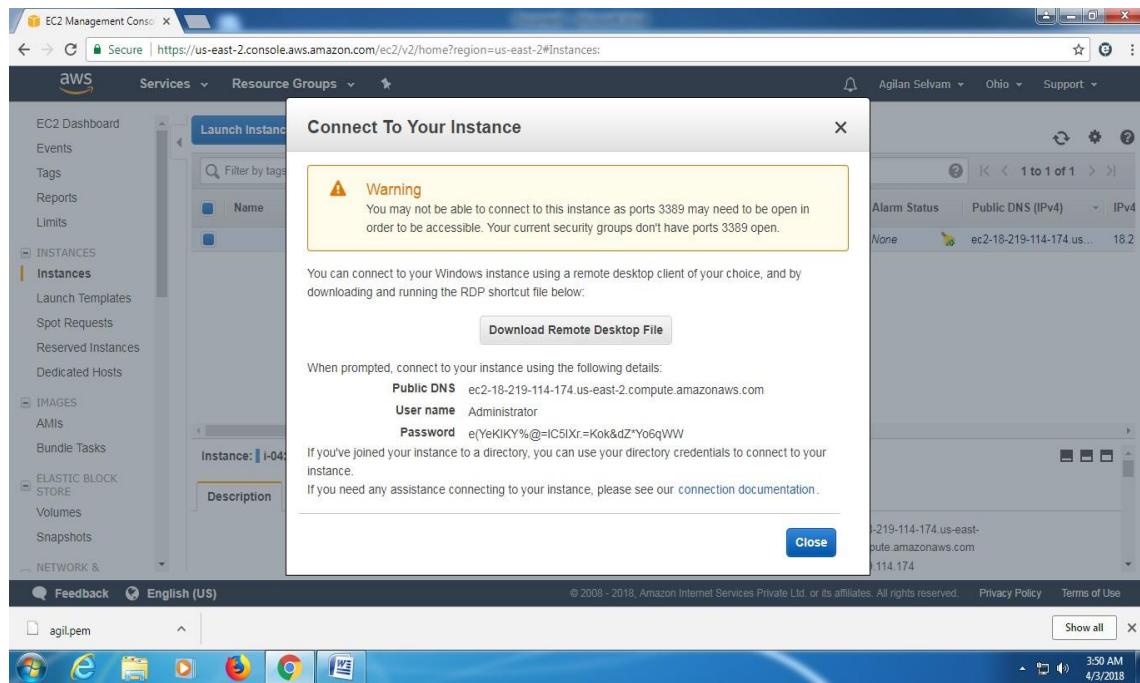


Click view launch log and check the status of the instance and click connectto connect to the instance.

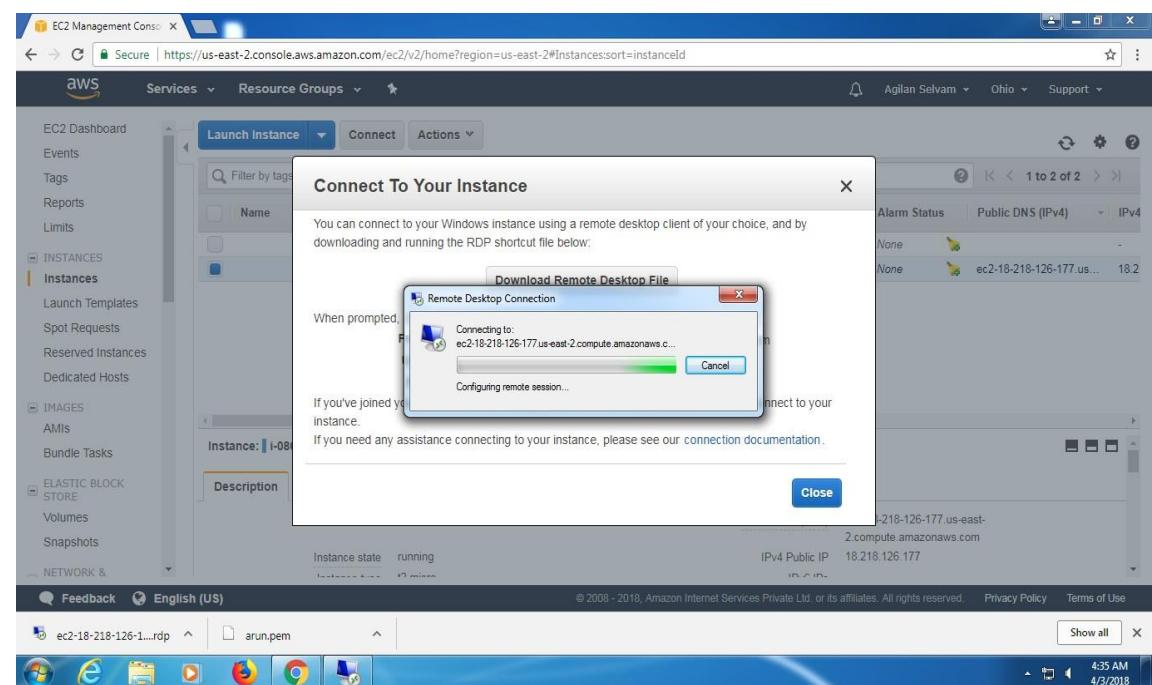


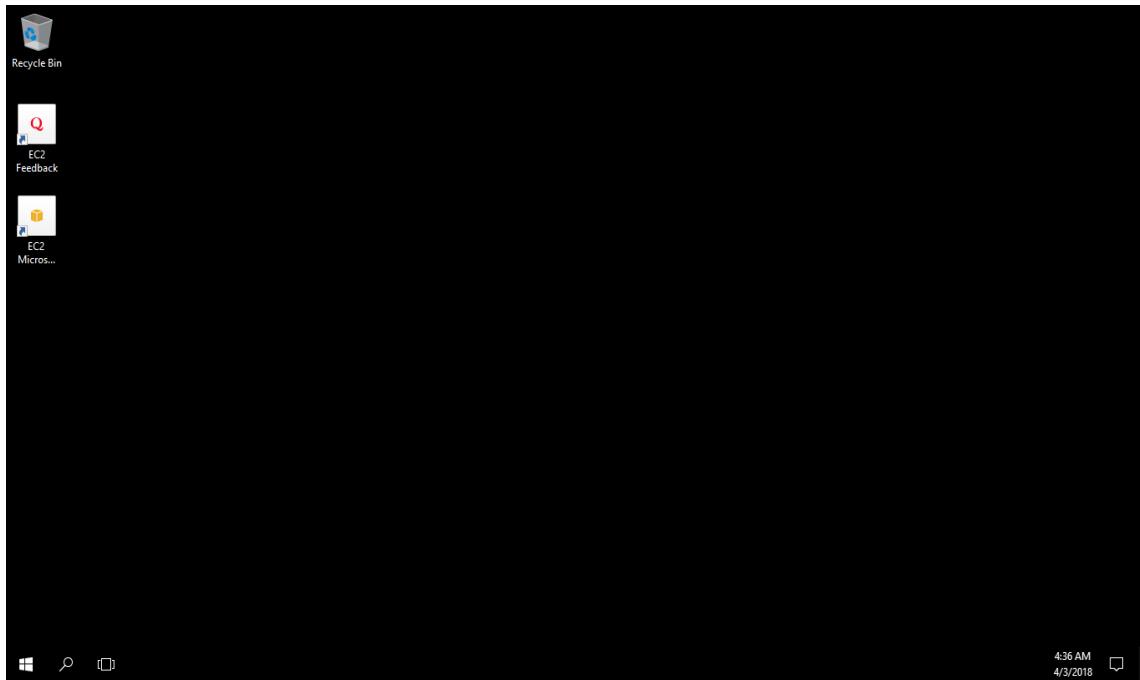
Click Get password and choose the file which saved locally earlier and click Decrypt password.





- ii. Click Download Remote Desktop File and open that file, thus the instance islaunched successfully.





DEPARTMENT OF CSE		
Preparation(Algorithm)	4	
Observation(Program)	4	
Results(Output)	4	
Interpretation(Validation)	4	
Viva-Voce	4	
Total	20	

RESULT:

EXP NO : 9

IBM ACCOUNT - IOT APPLICATION WITH NODE RED

DATE:

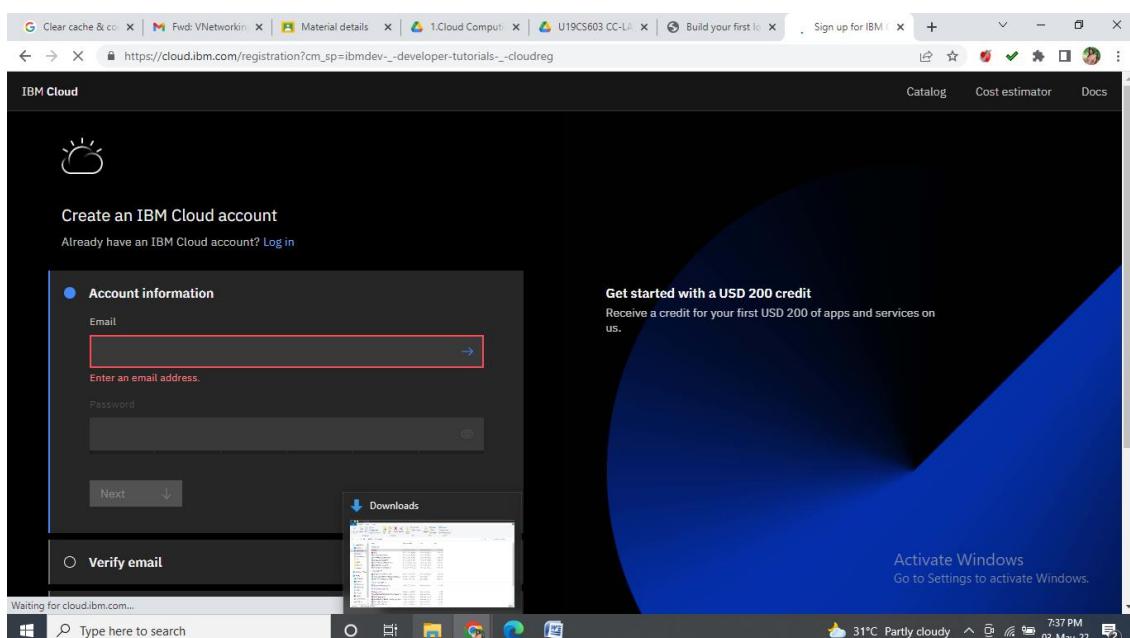
AIM:

To detect a covid 19 over a region using a machine learning techniques and algorithms.

PROCEDURE:

Step 1. Find the Node-RED Starter Kit in the IBM Cloud catalog

- Create a Node-RED starter application running in the IBM Cloud
- Secure the application
- Customize the Node-RED Starter Kit by adding additional nodes
- Signup for IBM cloud account



- Open the catalog and search for **node-red**.
- Click on the **Node-RED App** tile.

Step 2. Create your application

- On the *Create* tab, a randomly generated **App name** will be suggested. Either accept that default name or provide a unique name for your application. This will become part of the application URL.

Note: If the name is not unique, you will see an error message and you must enter a different name before you can continue.

- The Node-RED starter application requires an instance of the **Cloudant database service** with IBM Cloud IAM and Cloudant credentials to store your application flow configuration. Select the region the service should be created in and what pricing plan it should use.
- Click the **Create** button to continue. This will create your application and, if necessary a Cloudant database service instance, but it is not yet deployed to IBM Cloud.

Step 3. Enable the Continuous Delivery feature

At this point, you have created the application and the resources it requires, but you have not deployed it anywhere to run. This step shows how to setup the Continuous Delivery feature that will deploy your application into the **Code Engine** space of IBM Cloud.

1. On the next screen, click the **Deploy your app** button to enable the *Continuous Delivery* feature for your application.
2. On the next screen, click the **Code Engine** tile.
3. Scroll down after selecting the **Code Engine** tile. You will need to create an **IBM Cloud API** key to allow the deployment process to access your resources. Click the **New** button to create the key. A message dialog will appear. You can accept the default values and confirm to close the dialog.
4. Select the **Region** and **Container registry region**, to deploy your application to. This should match the region you created your Cloudant instance in.
5. Provide a unique **Project** name or accept the default 'project-name'

Click **Next** to continue.

6. Configure the **DevOps toolchain** by selecting the **region** it should be created in. Again, try to match the region you selected previously.

Click **Create**. This will take you back to the application details page. The application details page gives you a useful overview of your application including details of its external URL¹⁰ and the additional services it is connected to. Bookmark this page as you'll need to come back to it later.

- After a few moments, the **Deployment Automation** section will refresh with the details of your newly created Delivery Pipeline. The Status field of the pipeline will eventually show **In progress**. That means your application is being built and deployed.
- The Deploy stage will take a few minutes to complete. You can click on the ci-pipeline **Status** link to check the progress of the Delivery Pipeline. Eventually the Deploy stage will display a green checkmark and a **Success** message to show it has passed. This means your Node-RED starter application is now running.

Step 4. Open the Node-RED application

Now that you've deployed your Node-RED application, let's open it up! You may have to refresh your page.

On the application details page, you should now see the **App URL**, **Source** and **Deployment target** fields filled in.

The screenshot shows the IBM Cloud App Details page for an application named "node-red-starter-sample1".

- Deployment Automation:**
 - Name: node-red-starter-sample1
 - Location: Dallas
 - Tool integrations: Icons for Google Cloud, AWS Lambda, Docker, and GitHub.
- Delivery Pipelines:**
 - ci-pipeline: Success
 - pr-pipeline: No stages detected
- Details:**

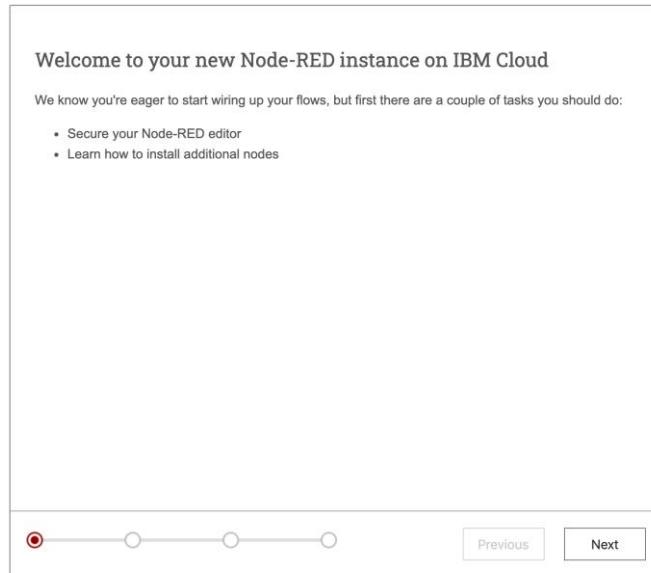
App URL	https://noderedstartersample1.dvgfmmubsee.us-south.codeengine.cloud.ibm.com/
Source	https://us-south.git.cloud.ibm.com/jawalicki/node-red-starter-sample1
Resource group	Default
Deployment target	Code Engine
Created	9/1/2021
- Services:**
 - Cloudant: Open dashboard, Documentation, API reference
 - Credentials: A dropdown menu.
- Actions:** A button labeled "Actions..." with a dropdown arrow.

Click on the **App URL** to open up your Node-RED application in a new browser tab.

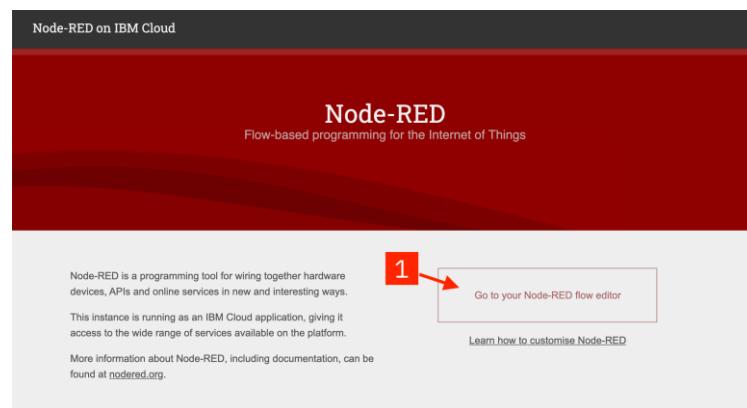
Step 5. Configure your Node-RED application

The first time you open your Node-RED app, you'll need to configure it and set up security.

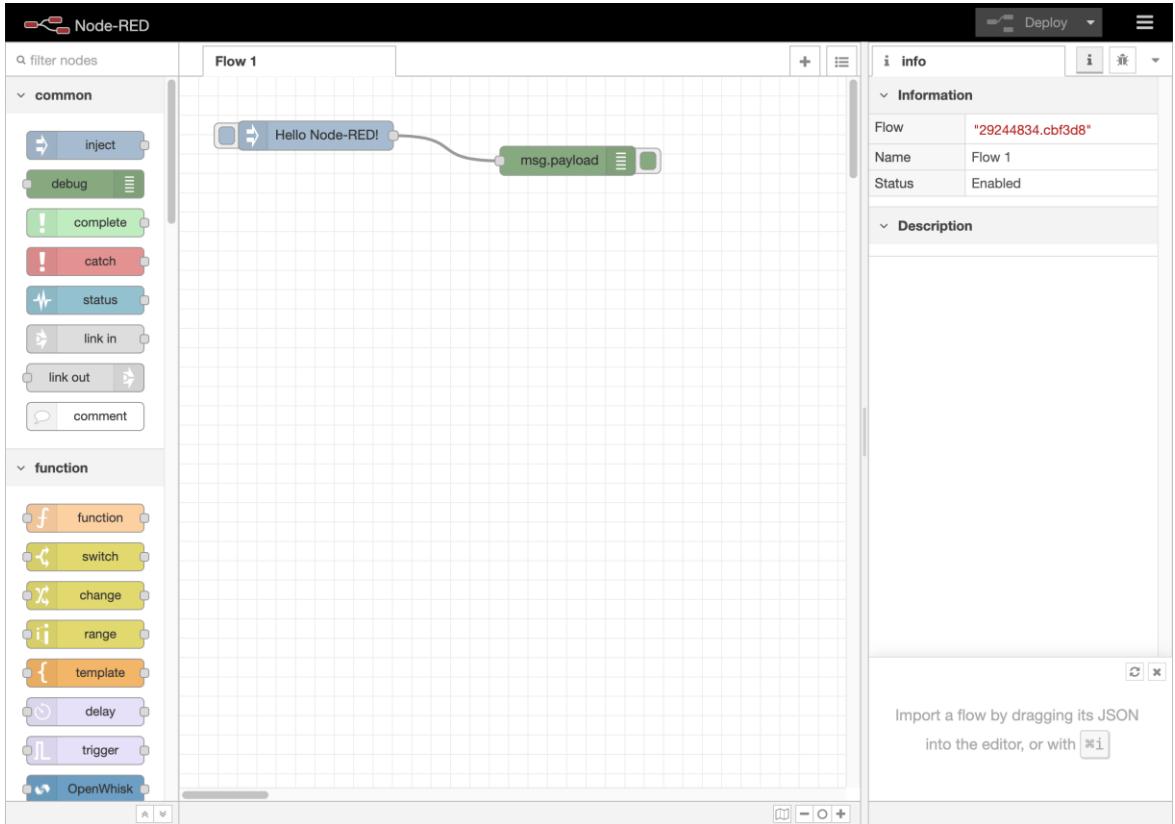
1. A new browser tab will open with the Node-RED start page.



2. On the initial screen, click **Next** to continue.
3. Secure your Node-RED editor by providing a **username** and **password**. If you need to change these at any point, you can either edit the values in the Cloudant database, or override them using *environment variables*. The documentation on nodered.org describes how to do this. Click **Next** to continue.
4. The final screen summarizes the options you've made and highlights the environment variables you can use to change the options in the future. Click **Finish** to proceed.
5. Node-RED will save your changes and then load the main application. From here you can click the **Go to your Node-RED flow editor** button to open the editor.



The Node-RED editor opens showing the default flow.



Step 6. Add extra nodes to your Node-RED palette

Node-RED provides the palette manager feature that allows you to install additional nodes directly from the browser-based editor. This is convenient for trying nodes out, but it can cause issues due to the limited memory of the default Node-RED starter application.

The recommended approach is to edit your application's package.json file to include the additional node modules and then redeploy the application.

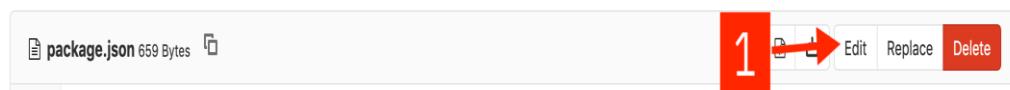
This step shows how to do that in order to add the [node-red-dashboard](#) module.

1. On your application's details page, click **Source url**. This will take you to a git repository where you can edit the application source code from your browser.
2. Scroll down the list of files and click on **package.json**. This file lists the module dependencies of your application.

The screenshot shows the IBM Cloud interface for a project named 'NodeREDSSLPB'. The left sidebar includes options like Project, Details, Activity, Releases, Repository, Issues (0), Merge Requests (0), Wiki, Snippets, and Settings. The right pane displays a list of files in the repository. A red arrow points to the 'package.json' file, which is highlighted with a red box containing the number '1'.

File	Action	Last Modified
DCO1.1.txt	clone from zip	3 weeks ago
Dockerfile	clone from zip	3 weeks ago
Dockerfile-tools	clone from zip	3 weeks ago
LICENSE	clone from zip	3 weeks ago
README.md	clone from zip	3 weeks ago
bluemix-settings.js	clone from zip	3 weeks ago
cli-config.yml	clone from zip	3 weeks ago
cloudantStorage.js	clone from zip	3 weeks ago
index.js	clone from zip	3 weeks ago
manifest.yml	clone from zip	3 weeks ago
package.json	clone from zip	3 weeks ago
run-debug	clone from zip	3 weeks ago
run-dev	clone from zip	3 weeks ago
service.yaml	clone from zip	3 weeks ago
README.md		

3. Click the **Edit** button



4. Add the following entry to the top of the dependencies section (1):
5. “node-red-dashboard”: “2.x”,

Add a **Commit message** (2) and click **Commit changes** (3)

Edit file

```

1 + {
2   "name": "node-red-app",
3   "version": "1.1.1",
4   "dependencies": {
5     "node-red-dashboard": "2.x",
6     "@cloudant/cloudant": "^4.1.1",
7     "appmetrics-dash": "^5.2.0",
8     "bcrypt": "3.0.7",
9     "body-parser": "1.x",
10    "cfenv": "1.2.2",
11    "express": "4.x",
12    "http-shutdown": "1.2.1",
13    "node-red": "1.x",
14    "node-red-node-cf-cloudant": "0.x",
15    "node-red-node-openwhisk": "0.x",
16    "node-red-node-watson": "0.x",
17    "node-red-nodes-cf-sqldb-dashdb": "0.x",
18    "ibm-cloud-env": "4.0"
19  },
20  "scripts": {
21    "start": "node --max-old-space-size=160 index.js --settings ./bluemix-settings.js -v"
22  },
23  "engines": {
24    "node": "12.x"
25  }
26}

```

Commit message: Add node-red-dashboard

Target Branch: master

Commit changes

- At this point, the Continuous Delivery pipeline will automatically run to build and deploy that change into your application. If you view the Delivery Pipeline you can watch its progress. The Build section shows you the last commit made and the Deploy section shows the progress of redeploying the application.

Status	All	Trigger	All	Created	Duration			
<input type="checkbox"/>	#	Status	Name	Pipeline	Trigger ⓘ	Created	Duration	
<input type="checkbox"/>	2		code-engine-cef2...	code-engine-pipel...		commit-push	September 23, 20...	52 seconds
<input type="checkbox"/>	1		code-engine-56c0...	code-engine-pipel...		manual-run	September 23, 20...	6 minutes 43 seco...

- Once the Deploy stage completes, your application will have restarted and now have the node-red-dashboard nodes preinstalled.

DEPARTMENT OF CSE		
Preparation(Algorithm)	4	
Observation(Program)	4	
Results(Output)	4	
Interpretation(Validation)	4	
Viva-Voce	4	
Total	20	

RESULT:

EXP NO 10

MICROSOFT AZURE ACCOUNT – BOT APPLICATION

DATE:

AIM

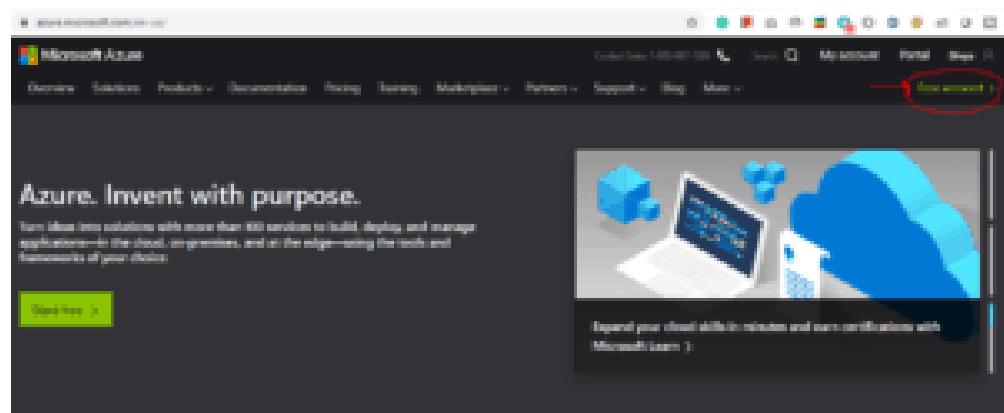
Create a free account in Microsoft Azure cloud and develop a bot for business applications

PROCEDURE

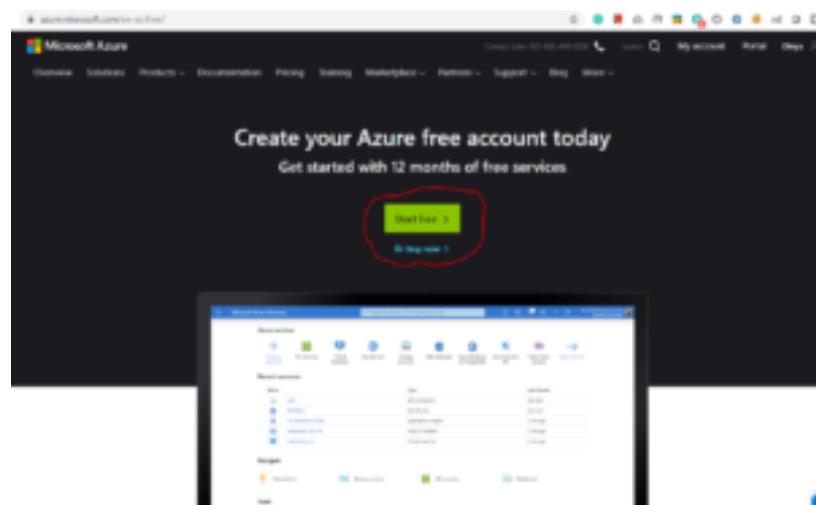
1. Go to the [Azure Home Page](https://azure.microsoft.com/en-us/)

<https://azure.microsoft.com/en-us/>

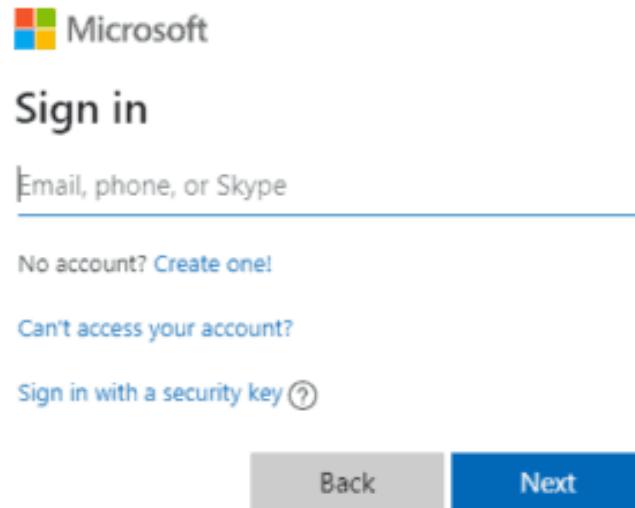
2. Click on **Free Azure Account** on the top right corner.



3. Click on **Start Free**.



4. Sign-in/Sign-up for a Microsoft account using an email address and password.



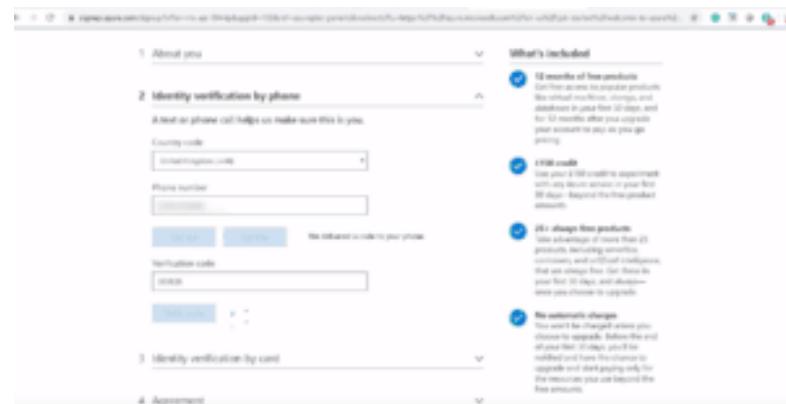
5. Enter your Country/Region and Date of Birth and click next.

6. Enter the **verification code** received on the email address and click next.

7. Type the captcha you see on your screen and click on next.

8. Redirected to the Azure Sign-up page. Enter your Region, Name, Phone number, Email address.

9. Verify your phone number by clicking Text Me or Call Me and enter the verification code received.



10. Enter the payment details. Make sure you have a **Master Card/American Express/ Visa Credit card** and international payments should be enabled.

The screenshot shows the Microsoft Azure sign-up page. On the left, there's a form for entering payment card information, including fields for cardholder name, card number, expiration date, and address details. On the right, there are three circular icons with text descriptions:

- No always-free products:** Take advantage of more than 25 products, including services, containers, and artificial intelligence, that are always free. Get these in your free 10 days, and always—once you choose to upgrade.
- 10 days—explore the free product amounts:**
- No automatic charges:** You won't be charged unless you choose to upgrade. Before the end of your first 10 days, you'll be notified and have the chance to upgrade and start paying only for the resources you use beyond the free amounts.

11. Check the Terms and Conditions and click Sign-up.

The screenshot shows the Microsoft Azure sign-up confirmation page. It features a blue header bar with the text "Follow these steps to get started. We ask for these details to protect your account and information. There are no upfront charges or fees." Below this, there are several sections:

- 1 About you**
- 2 Identity verification by phone**
- 3 Identity verification by card**
- 4 Agreement**

In the "Agreement" section, there are two checked checkboxes:

- I agree to the [Microsoft Terms of Use](#), [Privacy Statement](#), and [Cookie Statement](#).
- I want the information, tips, and offers from Microsoft to selected partners about Azure, including Azure Resources, Pricing updates, and other Microsoft products and services.

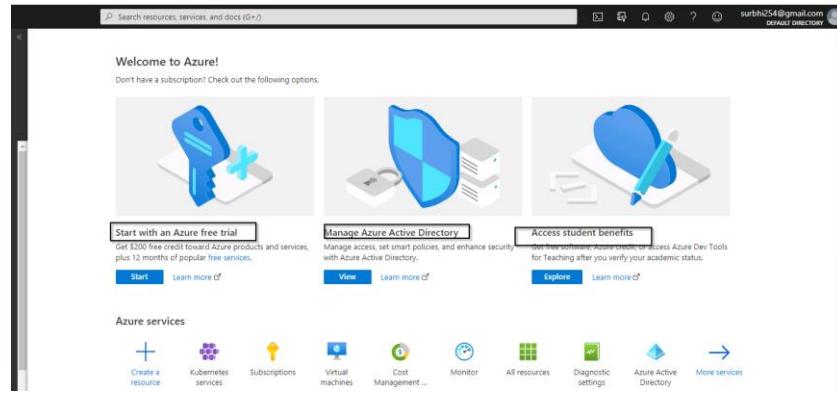
Below these checkboxes is a blue "Sign-up" button. To the right of the steps, there's a section titled "What's included" with four items, each preceded by a blue circular icon with a checkmark:

- 10 months of free products:** Get free access to popular products like virtual machines, storage, and databases in your free 10 days, and for 10 months after you upgrade your account to pay-as-you-go pricing.
- £100 credit:** Use your £100 credit to experiment with the Azure service in your free 10 days—explore the free product amounts.
- 10 days—explore the free product amounts:**
- No automatic charges:** You won't be charged unless you choose to upgrade. Before the end of your first 10 days, you'll be notified and have the chance to upgrade and start paying only for the resources you use beyond the free amounts.

12. Successfully created a Microsoft Azure free account and now have a lumpsum balance of **\$200**.

13. Click on **Portal** on the top right corner of the screen. You'll be redirected to the Azure portal.

14. If you have crossed the limit or time limit then you will get “your subscription is disabled and cannot perform operations until its re-enabled”



You can not create multiple Microsoft azure free accounts or after one account expires to another account for free credit of 200\$ using a single Credit card.

DEPARTMENT OF CSE		
Preparation(Algorithm)	4	
Observation(Program)	4	
Results(Output)	4	
Interpretation(Validation)	4	
Viva-Voce	4	
Total	20	

RESULT